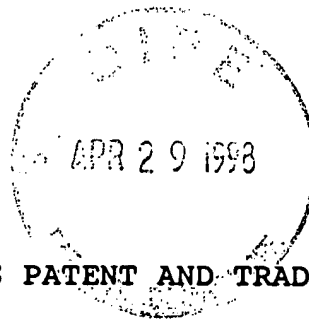


862.2136



Received 4-14-98  
OCT 15 1998  
TR

PATENT APPLICATION 9700

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of: )  
KOH I FUKUNAGA ET AL. ) Examiner: Not Yet Assigned  
Application No.: 09/025,133 ) Group Art Unit: 2782  
Filed: February 17, 1998 )  
For: DATA TRANSMISSION )  
APPARATUS, SYSTEM AND )  
METHOD, AND IMAGE )  
PROCESSING APPARATUS : April 28, 1998

Assistant Commissioner for Patents  
Washington, D.C. 20231

CLAIM TO PRIORITY

Sir:

Applicants hereby claim priority under the  
International Convention and all rights to which they are  
entitled under 35 U.S.C. § 119 based upon the following  
Japanese Priority Applications:

9-030982, filed February 14, 1997;

9-127644, filed May 16, 1997; and

9-127652, filed May 16, 1997.

Certified copies of the priority documents are  
enclosed.

RECEIVED  
OCT 15 1998  
11 30 57  
0093 2600

Applicants' undersigned attorney may be reached in our New York office by telephone at (212) 758-2400. All correspondence should continue to be directed to our below listed address.

Respectfully submitted,

Abigail Cousins  
Attorney for Applicants

Registration No. 29,292

FITZPATRICK, CELLA, HARPER & SCINTO  
277 Park Avenue  
New York, New York 10172  
Facsimile: (212) 758-2982

A:\862-2136.CTP\rmc

(translation of the front page of the priority document of  
Japanese Patent Application No.9-030982

PATENT OFFICE  
JAPANESE GOVERNMENT

This is to certify that the annexed is a true copy of the  
following application as filed with this Office.

Date of Application: February 14, 1997

Application Number : Patent Application 9-030982

Applicant(s) : Canon Kabushiki Kaisha

March 13, 1998

Commissioner,  
Patent Office

Hisamitsu Arai

Certification Number 10-3013998



日 本 国 特 許 庁

PATENT OFFICE  
JAPANESE GOVERNMENT

CFM 1206  
US. EP  
H.N. 09/025/13  
CAU 1182

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日

Date of Application:

1997年 2月14日

出 願 番 号

Application Number:

平成 9年特許願第030982号

出 願 人

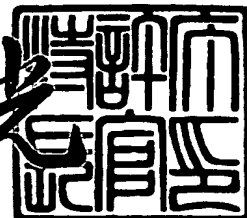
Applicant (s):

キヤノン株式会社

1998年 3月13日

特 許 庁 長 官  
Commissioner,  
Patent Office

荒井 寿光



出証番号 出証特平10-3013998

【書類名】 特許願

【整理番号】 3452032

【提出日】 平成 9年 2月14日

【あて先】 特許庁長官殿

【国際特許分類】 H04N 5/00

【発明の名称】 データ処理方法、データ処理装置、プリンタ及び記憶媒体

【請求項の数】 27

【発明者】

    【住所又は居所】 東京都大田区下丸子3丁目30番2号 キヤノン株式会社  
社内

    【氏名】 立山 二郎

【発明者】

    【住所又は居所】 東京都大田区下丸子3丁目30番2号 キヤノン株式会社  
社内

    【氏名】 福長 耕司

【発明者】

    【住所又は居所】 東京都大田区下丸子3丁目30番2号 キヤノン株式会社  
社内

    【氏名】 片野 清

【発明者】

    【住所又は居所】 東京都大田区下丸子3丁目30番2号 キヤノン株式会社  
社内

    【氏名】 小林 真琴

【発明者】

    【住所又は居所】 東京都大田区下丸子3丁目30番2号 キヤノン株式会社  
社内

    【氏名】 中村 敦

【発明者】

【住所又は居所】 東京都大田区下丸子3丁目30番2号 キヤノン株式会社  
社内

【氏名】 鈴木 尚久

【特許出願人】

【識別番号】 000001007

【氏名又は名称】 キヤノン株式会社

【代表者】 御手洗 富士夫

【代理人】

【識別番号】 100090273

【弁理士】

【氏名又は名称】 國分 孝悦

【電話番号】 03-3590-8901

【手数料の表示】

【予納台帳番号】 035493

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9117732

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 データ処理方法、データ処理装置、プリンタ及び記憶媒体

【特許請求の範囲】

【請求項1】 共通シリアスバスを介して複数種類のプリンタのプロトコルを切り換えるためのデータ処理方法であって、

前記プリンタのプロトコルの種類に係わらない初期プロトコルを実行し、

前記初期プロトコルに続いて前記複数種類のプリンタ固有のプロトコルを実行することを特徴とするデータ処理方法。

【請求項2】 前記共通シリアルバスは、IEEE1394規格に適合したバスであることを特徴とする請求項1記載のデータ処理方法。

【請求項3】 前記プリンタのプロトコルは、インクジェットプリンタ用のプロトコルであることを特徴とする請求項1記載のデータ処理方法。

【請求項4】 前記共通シリアルバスは、USB規格に適合したバスであることを特徴とする請求項1記載のデータ処理方法。

【請求項5】 前記初期プロトコルは、OSIモデルのデータリンク層よりも上位の層で実行されるプロトコルであることを特徴とする請求項1記載のデータ処理方法。

【請求項6】 前記プリンタ固有のプロトコル実行後に、プリントすべき画像データを伝送することを特徴とする請求項1記載のデータ処理方法。

【請求項7】 前記画像データは、撮像手段によって光電変換されたデータであることを特徴とする請求項6記載のデータ処理方法。

【請求項8】 共通シリアスバスを介して複数種類のプリンタのプロトコルを切り換えるためのデータ処理装置であって、

前記プリンタのプロトコルの種類に係わらない初期プロトコルを実行する第1の実行手段と、

前記初期プロトコルに続いて前記複数種類のプリンタ固有のプロトコルを実行する第2の実行手段とを備えることを特徴とするデータ処理装置。

【請求項9】 前記共通シリアルバスは、IEEE1394規格に適合したバスであることを特徴とする請求項8記載のデータ処理装置。

【請求項10】 前記プリンタのプロトコルは、インクジェットプリンタ用のプロトコルであることを特徴とする請求項8記載のデータ処理装置。

【請求項11】 前記共通シリアルバスは、USB規格に適合したバスであることを特徴とする請求項8記載のデータ処理装置。

【請求項12】 前記初期プロトコルは、OSIモデルのデータリンク層よりも上位の層で実行されるプロトコルであることを特徴とする請求項8記載のデータ処理装置。

【請求項13】 前記第2の実行手段は、前記プリンタ固有のプロトコル実行後に、プリントすべき画像データを伝送することを特徴とする請求項8記載のデータ処理装置。

【請求項14】 光電変換を行って得た前記画像データを前記第2の実行手段に与える撮像手段を備えることを特徴とする請求項13記載のデータ処理装置。

【請求項15】 共通シリアルバスを介して複数種類のプリンタのプロトコルをホストに対応して切り換えるプリンタであって、

前記プロトコルの種類に係わらない初期プロトコルを実行する実行手段と、  
前記初期プロトコルに続いてプリンタ固有のプロトコルを実行した後、印字データを受信する受信手段とを備えることを特徴とするプリンタ。

【請求項16】 前記共通シリアルバスは、IEEE1394規格に適合したバスであることを特徴とする請求項15記載のプリンタ。

【請求項17】 前記プリンタのプロトコルは、インクジェットプリンタ用のプロトコルであることを特徴とする請求項15記載のプリンタ。

【請求項18】 前記共通シリアルバスは、USB規格に適合したバスであることを特徴とする請求項15記載のプリンタ。

【請求項19】 前記初期プロトコルは、OSIモデルのデータリンク層よりも上位の層で実行されるプロトコルであることを特徴とする請求項15記載のプリンタ。

【請求項20】 光電変換を行って得た画像データを前記印字データとして前記受信手段に送信する撮像手段を備えることを特徴とする請求項15記載のデータ処理装置。



ータ処理装置。

【請求項21】 共通シリアスバスを介して複数種類のプリンタのプロトコルを切り換えるためのデータ処理を実行するためのステップが格納された記憶媒体であって、

前記プリンタのプロトコルの種類に係わらない初期プロトコルを実行する第1の実行ステップと、

前記初期プロトコルに続いて前記複数種類のプリンタ固有のプロトコルを実行する第2の実行ステップとを、コンピュータが読出可能に格納したことを特徴とする記憶媒体。

【請求項22】 前記共通シリアルバスは、IEEE1394規格に適合したバスであることを特徴とする請求項21記載の記憶媒体。

【請求項23】 前記プリンタのプロトコルは、インクジェットプリンタ用のプロトコルであることを特徴とする請求項21記載の記憶媒体。

【請求項24】 前記共通シリアルバスは、USB規格に適合したバスであることを特徴とする請求項21記載の記憶媒体。

【請求項25】 前記初期プロトコルは、OSIモデルのデータリンク層よりも上位の層で実行されるプロトコルであることを特徴とする請求項21記載の記憶媒体。

【請求項26】 前記第2の実行ステップは、前記プリンタ固有のプロトコル実行後に、プリントすべき画像データを伝送するステップを含むことを特徴とする請求項21記載の記憶媒体。

【請求項27】 前記画像データは、撮像手段によって光電変換されたデータであることを特徴とする請求項26記載の記憶媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、共通シリアスバスを介して、複数種類のプリンタのプロトコルを切り換えるためのデータ処理方法、データ処理装置、プリンタ及び記憶媒体に関するものである。

【0002】

【従来の技術】

従来より、シリアスバスを介してプリンタにデータを送出するシステムとして、様々な種類のシステムが知られている。

【0003】

例えば、SCSI (Small Computer System Interface)、セントロニクス等、一般に広く用いられるようになったデファクトスタンダードのインターフェースを用いて、コンピュータからプリンタにデータを出力する技術が知られている。

【0004】

【発明が解決しようとする課題】

しかしながら、シリアスバスを用いてプリンタデータを送出する従来のプリンタプロトコルは、プリンタメーカー固有の1つに限られていた。

したがって、拡張性に欠ける、という問題が生じていた。

特に、様々な種類の機器を接続するインターフェース、例えば、IEEE 1394のようなインターフェースを用いて、プリントデータを出力する際には、かかる拡張性に欠けるという問題点は解決すべき大きな課題であった。

【0005】

そこで、本発明は、上記の欠点を除去するために成されたもので、システムバスを介してプリントデータを送出する際に、拡張性の高いデータ処理方法、データ処理装置、プリンタ及び記憶媒体を提供することを目的とする。

また、本発明は、IEEE 1394規格に好適なデータ処理方法、データ処理装置、プリンタ及び記憶媒体を提供することを目的とする。

また、本発明は、コンピュータを介することなく、画像データ出力デバイスから直接プリンタを接続するのに好適なデータ処理方法、データ処理装置、プリンタ及び記憶媒体を提供することを目的とする。

【0006】

【課題を解決するための手段】

本発明は、共通シリアスバスを介して複数種類のプリンタのプロトコルを切り

換えるためのデータ処理方法であって、前記プリンタのプロトコルの種類に係わらない初期プロトコルを実行し、前記初期プロトコルに続いて前記複数種類のプリンタ固有のプロトコルを実行することを特徴とする。

また、前記共通シリアルバスは、IEEE 1394規格に適合したバスであることを特徴とする。

また、前記プリンタのプロトコルは、インクジェットプリンタ用のプロトコルであることを特徴とする。

また、前記共通シリアルバスは、USB規格に適合したバスであることを特徴とする。

また、前記初期プロトコルは、OSIモデルのデータリンク層よりも上位の層で実行されるプロトコルであることを特徴とする。

また、前記プリンタ固有のプロトコル実行後に、プリントすべき画像データを伝送することを特徴とする。

また、前記画像データは、撮像手段によって光電変換されたデータであることを特徴とする。

本発明は、共通シリアルバスを介して複数種類のプリンタのプロトコルを切り換えるためのデータ処理装置であって、前記プリンタのプロトコルの種類に係わらない初期プロトコルを実行する第1の実行手段と、前記初期プロトコルに続いて前記複数種類のプリンタ固有のプロトコルを実行する第2の実行手段とを備えることを特徴とする。

また、前記共通シリアルバスは、IEEE 1394規格に適合したバスであることを特徴とする。

また、前記プリンタのプロトコルは、インクジェットプリンタ用のプロトコルであることを特徴とする。

また、前記共通シリアルバスは、USB規格に適合したバスであることを特徴とする。

また、前記初期プロトコルは、OSIモデルのデータリンク層よりも上位の層で実行されるプロトコルであることを特徴とする。

また、前記第2の実行手段は、前記プリンタ固有のプロトコル実行後に、プリ

ントすべき画像データを伝送することを特徴とする。

また、光電変換を行って得た前記画像データを前記第2の実行手段に与える撮像手段を備えることを特徴とする。

本発明は、共通シリアスバスを介して複数種類のプリンタのプロトコルをホストに対応して切り換えるプリンタであって、前記プロトコルの種類に係わらない初期プロトコルを実行する実行手段と、前記初期プロトコルに続いてプリンタ固有のプロトコルを実行した後、印字データを受信する受信手段とを備えることを特徴とする。

また、前記共通シリアルバスは、IEEE1394規格に適合したバスであることを特徴とする。

また、前記プリンタのプロトコルは、インクジェットプリンタ用のプロトコルであることを特徴とする。

また、前記共通シリアルバスは、USB規格に適合したバスであることを特徴とする。

また、前記初期プロトコルは、OSIモデルのデータリンク層よりも上位の層で実行されるプロトコルであることを特徴とする。

また、光電変換を行って得た画像データを前記印字データとして前記受信手段に送信する撮像手段を備えることを特徴とする。

本発明は、共通シリアスバスを介して複数種類のプリンタのプロトコルを切り換えるためのデータ処理を実行するためのステップが格納された記憶媒体であって、前記プリンタのプロトコルの種類に係わらない初期プロトコルを実行する第1の実行ステップと、前記初期プロトコルに続いて前記複数種類のプリンタ固有のプロトコルを実行する第2の実行ステップとを、コンピュータが読出可能に格納したことを特徴とする。

また、前記共通シリアルバスは、IEEE1394規格に適合したバスであることを特徴とする。

また、前記プリンタのプロトコルは、インクジェットプリンタ用のプロトコルであることを特徴とする。

また、前記共通シリアルバスは、USB規格に適合したバスであることを特徴

とする。

また、前記初期プロトコルは、OSIモデルのデータリンク層よりも上位の層で実行されるプロトコルであることを特徴とする。

また、前記第2の実行ステップは、前記プリンタ固有のプロトコル実行後に、プリントすべき画像データを伝送するステップを含むことを特徴とする。

また、前記画像データは、撮像手段によって光電変換されたデータであることを特徴とする。

【0007】

【発明の実施の形態】

【0008】

以下、本発明の実施の形態について図面を用いて説明する。

【0009】

ここで、まず、以下に説明する第1及び第2の実施の形態では、各機器間を接続するデジタルI/Fとして、IEEE1394シリアルバスを用いているため、IEEE1394シリアルバスについて、予め説明する。

【0010】

民生用デジタルVCRやDVDプレーヤの登場に伴って、ビデオデータやオーディオデータなどのリアルタイムで、かつ高情報量のデータ転送のサポートが必要になっている。こういったビデオデータやオーディオデータをリアルタイムで転送し、パソコン(PC)に取り込んだり、またはその他のデジタル機器に転送を行なうには、必要な転送機能を備えた高速データ転送可能なインタフェースが必要になってくるものであり、そういった観点から開発されたインタフェースが、IEEE1394-1995 (High Performance Serial Bus) (以下、単に1394シリアルバスとも言う)である。

【0011】

図1に1394シリアルバスを用いて構成されるネットワーク・システムの例を示す。

【0012】

このシステムは、機器A,B,C,D,E,F,G,Hを備えており、A-B間、A-C間、B-D

間、D-E 間、C-F 間、C-G 間、及びC-H 間をそれぞれ1394シリアルバスのツイスト・ペア・ケーブルで接続されている。この機器A～Hは、例えばパソコン、デジタルVTR、DVD、デジタルカメラ、ハードディスク、モニタ、チューナー、モニター等である。

【0013】

各機器間の接続方式は、ディジーチェーン方式とノード分岐方式とを混在可能としたものであり、自由度の高い接続が可能である。

【0014】

また、各機器は各自固有のIDを有し、それぞれが認識し合うことによって1394シリアルバスで接続された範囲において、1つのネットワークを構成している。各デジタル機器間をそれぞれ1本の1394シリアルバスケーブルで順次接続するだけで、それぞれの機器が中継の役割を行い、全体として1つのネットワークを構成するものである。また、1394シリアルバス、Plug&Play機能でケーブルを機器に接続した時点で自動で機器の認識や接続状況などを認識する機能を有している。

【0015】

また、図1に示したようなシステムにおいて、ネットワークからある機器が削除されたり、または新たに追加されたときなど、自動的にバスリセットを行い、それまでのネットワーク構成をリセットしてから、新たなネットワークの再構築を行なう。この機能によって、その時々ネットワークの構成を常時設定、認識することができる。

【0016】

またデータ転送速度は、100 / 200 / 400 Mbpsと備えており、上位の転送速度を持つ機器が下位の転送速度をサポートし、互換をとるようになっている。

【0017】

データ転送モードとしては、コントロール信号などの非同期データ（Asynchronousデータ：以下Asyncデータ）を転送するAsynchronous転送モードとリアルタイムなビデオデータやオーディオデータ等の同期データ（Isochronousデータ：以下Isoデータ）を転送するIsochronous転送モードがある。このAsyncデータ

とIso データは各サイクル（通常1サイクル125  $\mu$ S）の中において、サイクル開始を示すサイクル・スタート・パケット（CSP）を転送した後、Iso データの転送をAsync データより優先しつつサイクル内で混在して転送される。

## 【0018】

つぎに、図2に1394シリアルバスの構成要素を示す。

## 【0019】

1394シリアルバスは全体としてレイヤ（階層）構造で構成されている。図2に示したように、1394シリアルバスのケーブルとコネクタが接続されるコネクタポートがあり、その上にハードウェアとしてフィジカル・レイヤとリンク・レイヤを位置づけしている。

## 【0020】

ハードウェア部は実質的なインターフェイスチップの部分であり、そのうちフィジカル・レイヤは符号化やコネクタ関連の制御等を行い、リンク・レイヤはパケット転送やサイクルタイムの制御等を行なう。

## 【0021】

ファームウェア部のトランザクション・レイヤは、転送（トランザクション）すべきデータの管理を行ない、Read、Write、Lockの命令を出す。マネージメント・レイヤは、接続されている各機器の接続状況やIDの管理を行ない、ネットワークの構成を管理する部分である。

## 【0022】

このハードウェアとファームウェアまでが実質上の1394シリアルバスの構成である。

## 【0023】

またソフトウェア部のアプリケーション・レイヤは使うソフトによって異なり、インタフェース上にどのようにデータをのせるか規定する部分であり、プリンタやAVCプロトコルなどが規定されている。

## 【0024】

以上が1394シリアルバスの構成である。

## 【0025】

つぎに、図3に1394シリアルバスにおけるアドレス空間の図を示す。

【0026】

1394シリアルバスに接続された各機器（ノード）には必ず各ノード固有の、64ビットアドレスを持たせておく。そしてこのアドレスをROMに格納しておくことで、自分や相手のノードアドレスを常時認識でき、相手を指定した通信も行なえる。

【0027】

1394シリアルバスのアドレッシングは、IEEE1212規格に準じた方式であり、アドレス設定は、最初の10bitがバスの番号の指定用に、次の6bitがノードID番号の指定用に使われる。残りの48bitが機器に与えられたアドレス幅になり、それぞれ固有のアドレス空間として使用できる。最後の28bitは固有データの領域として、各機器の識別や使用条件の指定の情報などを格納する。

【0028】

以上が1394シリアルバスの技術の概要である。

【0029】

つぎに、1394シリアルバスの特徴といえる技術の部分を、より詳細に説明する。

【0030】

1394シリアルバスの電氣的仕様について説明する。

【0031】

図4に1394シリアルバス・ケーブルの断面図を示す。

【0032】

1394シリアルバスでは接続ケーブル内に6ピン、即ち2組のツイストペア信号線の他に、電源ラインを設けている。これによって、電源を持たない機器や、故障により電圧低下した機器等にも電力の供給が可能になっている。

【0033】

電源線内を流れる電源の電圧は8～40V、電流は最大電流DC1.5Aと規定されている。

【0034】



なお、DVケーブルと呼ばれる規格では電源を省いた4ピンで構成されている。

【0035】

DS-Link符号化について説明する。

【0036】

1394シリアルバスで採用されている、データ転送フォーマットのDS-Link符号化方式を説明するための図を図5に示す。

【0037】

1394シリアルバスでは、DS-Link (Data/Strobe Link) 符号化方式が採用されている。このDS-Link符号化方式は、高速なシリアルデータ通信に適しており、その構成は、2本の信号線を必要とする。より対線のうち1本に主となるデータを送り、他方のより対線にはストロブ信号を送る構成になっている。受信側では、この通信されるデータと、ストロブとの排他的論理和をとることによってクロックを再現する。

【0038】

このDS-Link符号化方式を用いるメリットとして、8/10B変換に比べて転送効率が高いこと、PLL回路が不要となるのでコントローラLSIの回路規模を小さくできること、更には、転送すべきデータが無いときにアイドル状態であることを示す情報を送る必要が無いので、各機器のトランシーバ回路をスリープ状態にすることができることによって、消費電力の低減が図れる、などが挙げられる。

【0039】

バスリセットのシーケンスについて説明する。

【0040】

1394シリアルバスでは、接続されている各機器(ノード)にはノードIDが与えられ、ネットワーク構成として認識されている。

【0041】

このネットワーク構成に変化があったとき、例えばノードの挿抜や電源のON/OFFなどによるノード数の増減などによって変化が生じて、新たなネットワーク

構成を認識する必要があるとき、変化を検知した各ノードはバス上にバスリセット信号を送信して、新たなネットワーク構成を認識するモードに入る。

【0042】

このときの変化の検知方法は、1394ポート基盤上でのバイアス電圧の変化を検知することによって行われる。

【0043】

あるノードからバスリセット信号が伝達されて、各ノードのフィジカルレイヤはこのバスリセット信号を受けると同時にリンクレイヤにバスリセットの発生を伝達し、かつ他のノードにバスリセット信号を伝達する。最終的にすべてのノードがバスリセット信号を検知した後、バスリセットが起動となる。

【0044】

バスリセットは、先に述べたようなケーブル抜挿や、ネットワーク異常等によるハード検出による起動と、プロトコルからのホスト制御などによってフィジカルレイヤに直接命令を出すことによっても起動する。

また、バスリセットが起動するとデータ転送は一時中断され、この間のデータ転送は待たされ、終了後、新しいネットワーク構成のもとで再開される。

【0045】

以上がバスリセットのシーケンスである。

【0046】

ノードID決定のシーケンスについて説明する。

【0047】

バスリセットの後、各ノードは新しいネットワーク構成を構築するために、各ノードにIDを与える動作に入る。このときの、バスリセットからノードID決定までの一般的なシーケンスを図6、7、8のフローチャートを用いて説明する。

【0048】

図6のフローチャートは、バスリセットの発生からノードIDが決定し、データ転送が行えるようになるまでの、一連のバスの作業を示してある。

【0049】

まず、ステップS101として、ネットワーク内にバスリセットが発生することを常時監視していて、ここでノードの電源ON/OFFなどでバスリセットが発生するとステップS102に移る。

【0050】

ステップS102では、ネットワークがリセットされた状態から、新たなネットワークの接続状況を知るために、直接接続されている各ノード間において親子関係の宣言がなされる。ステップS103として、すべてのノード間で親子関係が決定すると、ステップS104として一つのルートが決定する。すべてのノード間で親子関係が決定するまで、ステップS102の親子関係の宣言をおこない、またルートも決定されない。

【0051】

ステップS104でルートが決定されると、次はステップS105として、各ノードにIDを与えるノードIDの設定作業が行われる。所定のノード順序で、ノードIDの設定が行われ、すべてのノードにIDが与えられるまで繰り返し設定作業が行われ、最終的にステップS106としてすべてのノードにIDを設定し終わったら、新しいネットワーク構成がすべてのノードにおいて認識されたので、ステップS107としてノード間のデータ転送が行える状態となり、データ転送が開始される。

【0052】

このステップS107の状態になると、再びバスリセットが発生するのを監視するモードに入り、バスリセットが発生したらステップS101からステップS106までの設定作業が繰り返し行われる。

【0053】

以上が、図6のフローチャートの説明であるが、図6のフローチャートのバスリセットからルート決定までの部分と、ルート決定後からID設定終了までの手順をより詳しくフローチャート図に表したものをそれぞれ、図7、図8に示す。

【0054】

まず、図7のフローチャートの説明を行う。

【0055】

ステップS201としてバスリセットが発生すると、ネットワーク構成は一旦リセットされる。なお、ステップS201としてバスリセットが発生するのを常に監視している。

## 【0056】

次に、ステップS202として、リセットされたネットワークの接続状況を再認識する作業の第一段階として、各機器にリーフ（ノード）であることを示すフラグを立てておく。さらに、ステップS203として各機器が自分の持つポートがいくつ他ノードと接続されているのかを調べる。

## 【0057】

ステップS204のポート数の結果に応じて、これから親子関係の宣言を始めていくために、未定義（親子関係が決定されてない）ポートの数を調べる。バスリセットの直後はポート数＝未定義ポート数であるが、親子関係が決定されていくにしたがって、ステップS204で検知する未定義ポートの数は変化していくものである。

## 【0058】

まず、バスリセットの直後、はじめに親子関係の宣言を行えるのはリーフに限られている。リーフであるというのはステップS203のポート数の確認で知ることができる。リーフは、ステップS205として、自分に接続されているノードに対して、「自分は子、相手は親」と宣言し動作を終了する。

## 【0059】

ステップS203でポート数が複数ありブランチと認識したノードは、バスリセットの直後はステップS204で未定義ポート数 $>1$ ということなので、ステップS206へと移り、まずブランチというフラグが立てられ、ステップS207でリーフからの親子関係宣言で「親」の受付をするために待つ。

## 【0060】

リーフが親子関係の宣言を行い、ステップS207でそれを受けたブランチは適宜ステップS204の未定義ポート数の確認を行い、未定義ポート数が1になっていれば残っているポートに接続されているノードに対して、ステップS205の「自分が子」の宣言をすることが可能になる。2度目以降、ステップS20

4で未定義ポート数を確認しても2以上あるブランチに対しては、再度ステップS207でリーフ又は他のブランチからの「親」の受付をするために待つ。

【0061】

最終的に、いずれか1つのブランチ、又は例外的にリーフ（子宣言を行えるのにすばやく動作しなかった為）がステップS204の未定義ポート数の結果としてゼロになったら、これにてネットワーク全体の親子関係の宣言が終了したものであり、未定義ポート数がゼロ（すべて親のポートとして決定）になった唯一のノードはステップS208としてルートのフラグが立てられ、ステップS209としてルートとしての認識がなされる。

【0062】

このようにして、図7に示したバスリセットから、ネットワーク内すべてのノード間における親子関係の宣言までが終了する。

【0063】

つぎに、図8のフローチャートについて説明する。

【0064】

まず、図8までのシーケンスでリーフ、ブランチ、ルートという各ノードのフラグの情報が設定されているので、これを元にして、ステップS301でそれぞれ分類する。

【0065】

各ノードにIDを与える作業として、最初にIDの設定を行うことができるのはリーフからである。リーフ→ブランチ→ルートの順で若い番号（ノード番号=0～）からIDの設定がなされていく。

【0066】

ステップS302としてネットワーク内に存在するリーフの数N（Nは自然数）を設定する。

【0067】

この後、ステップS303として各自リーフがルートに対して、IDを与えるように要求する。この要求が複数ある場合には、ルートはステップS304としてアービトレーションを行い、ステップS305として勝ったノード1つにID

番号を与え、負けたノードには失敗の結果通知を行う。

【0068】

ステップS306としてID取得が失敗に終わったリーフは、再度ID要求を出し、同様の作業を繰り返す。IDを取得できたリーフからステップS307として、そのノードのID情報をブロードキャストで全ノードに転送する。1ノードID情報のブロードキャストが終わると、ステップS308として残りのリーフの数が1つ減らされる。

【0069】

ここで、ステップS309として、この残りのリーフの数が1以上ある時はステップS303のID要求の作業からを繰り返し行い、最終的にすべてのリーフがID情報をブロードキャストすると、ステップS309が $N=0$ となり、次はブランチのID設定に移る。ブランチのID設定もリーフの時と同様に行われる。

【0070】

まず、ステップS310としてネットワーク内に存在するブランチの数 $M$  ( $M$ は自然数)を設定する。

【0071】

この後、ステップS311として各自ブランチがルートに対して、IDを与えるように要求する。これに対してルートは、ステップS312としてアービトレーションを行い、勝ったブランチから順にリーフに与え終った次の若い番号から与えていく。

【0072】

ステップS313として、ルートは要求を出したブランチにID情報又は失敗結果を通知し、ステップS314としてID取得が失敗に終わったブランチは、再度ID要求を出し、同様の作業を繰り返す。

【0073】

IDを取得できたブランチからステップS315として、そのノードのID情報をブロードキャストで全ノードに転送する。

【0074】

1ノードID情報のブロードキャストが終わると、ステップS316として残

りのブランチの数が1つ減らされる。

【0075】

ここで、ステップS317として、この残りのブランチの数が1以上ある時はステップS311のID要求の作業からを繰り返し、最終的にすべてのブランチがID情報をブロードキャストするまで行われる。すべてのブランチがノードIDを取得すると、ステップS317はM=0となり、ブランチのID取得モードも終了する。

【0076】

ここまで終了すると、最終的にID情報を取得していないノードはルートのみなので、ステップS318として与えていない番号で最も若い番号を自分のID番号と設定し、ステップS319としてルートのID情報をブロードキャストする。

【0077】

以上で、図8に示したように、親子関係が決定した後から、すべてのノードのIDが設定されるまでの手順が終了する。

【0078】

次に、一例として図9に示した実際のネットワークにおける動作を図9を参照しながら説明する。

【0079】

図9の説明として、(ルート)ノードBの下位にはノードAとノードCが直接接続されており、更にノードCの下位にはノードDが直接接続されており、更にノードDの下位にはノードEとノードFが直接接続された階層構造になっている。この、階層構造やルートノード、ノードIDを決定する手順を以下で説明する。

【0080】

バスリセットがされた後、まず各ノードの接続状況を認識するために、各ノードの直接接続されているポート間において、親子関係の宣言がなされる。この親子とは親側が階層構造で上位となり、子側が下位となるということができる。

【0081】

図9ではバスリセットの後、最初に親子関係の宣言を行なったのはノードAで

ある。基本的にノードの1つのポートにのみ接続があるノード（リーフと呼ぶ）から親子関係の宣言を行なうことができる。これは自分には1ポートの接続のみということをもとに知ることができるので、これによってネットワークの端であることを認識し、その中で早く動作を行なったノードから親子関係が決定されていく。こうして親子関係の宣言を行なった側（A-B間ではノードA）のポートが子と設定され、相手側（ノードB）のポートが親と設定される。こうして、ノードA-B間では子-親、ノードE-D間で子-親、ノードF-D間で子-親と決定される。

【0082】

さらに1階層あがって、今度は複数個接続ポートを持つノード（ブランチと呼ぶ）のうち、他ノードからの親子関係の宣言を受けたものから順次、更に上位に親子関係の宣言を行なっていく。図12ではまずノードDがD-E間、D-F間と親子関係が決定した後、ノードCに対する親子関係の宣言を行っており、その結果ノードD-C間で子-親と決定している。

【0083】

ノードDからの親子関係の宣言を受けたノードCは、もう一つのポートに接続されているノードBに対して親子関係の宣言を行なっている。これによってノードC-B間で子-親と決定している。

【0084】

このようにして、図9のような階層構造が構成され、最終的に接続されているすべてのポートにおいて親となったノードBが、ルートノードと決定された。

ルートは1つのネットワーク構成中に一つしか存在しないものである。

【0085】

なお、この図9においてノードBがルートノードと決定されたが、これはノードAから親子関係宣言を受けたノードBが、他のノードに対して親子関係宣言を早いタイミングで行なっていれば、ルートノードは他ノードに移っていたこともあり得る。すなわち、伝達されるタイミングによってはどのノードもルートノードとなる可能性があり、同じネットワーク構成でもルートノードは一定とは限らない。



【0086】

ルートノードが決定すると、次は各ノードIDを決定するモードに入る。ここではすべてのノードが、決定した自分のノードIDを他のすべてのノードに通知する（ブロードキャスト機能）。

【0087】

自己ID情報は、自分のノード番号、接続されている位置の情報、持っているポートの数、接続のあるポートの数、各ポートの親子関係の情報等を含んでいる。

【0088】

ノードID番号の割り振りの手順としては、まず1つのポートにのみ接続があるノード（リーフ）から起動することができ、この中から順にノード番号=0、1、2、・・・と割り当てられる。

【0089】

ノードIDを手にしたノードは、ノード番号を含む情報をブロードキャストで各ノードに送信する。これによって、そのID番号は「割り当て済み」であることが認識される。

【0090】

すべてのリーフが自己ノードIDを取得し終ると、次はブランチへ移りリーフに引き続いたノードID番号が各ノードに割り当てられる。リーフと同様に、ノードID番号が割り当てられたブランチから順次ノードID情報をブロードキャストし、最後にルートノードが自己ID情報をブロードキャストする。すなわち、常にルートは最大のノードID番号を所有するものである。

【0091】

以上のようにして、階層構造全体のノードIDの割り当てが終わり、ネットワーク構成が再構築され、バスの初期化作業が完了する。

【0092】

アービトレーションについて説明する。

【0093】

1394シリアルバスでは、データ転送に先立って必ずバス使用权のアービトレーション（調停）を行なう。1394シリアルバスは個別に接続された各機器

が、転送された信号をそれぞれ中継することによって、ネットワーク内すべての機器に同信号を伝えるように、論理的なバス型ネットワークであるので、パケットの衝突を防ぐ意味でアービトレーションは必要である。これによってある時間には、たった一つのノードのみ転送を行なうことができる。

【0094】

アービトレーションを説明するための図として図10（a）にバス使用要求の図（b）にバス使用許可の図を示し、以下これを用いて説明する。

【0095】

アービトレーションが始まると、1つもしくは複数のノードが親ノードに向かって、それぞれバス使用権の要求を発する。図10（a）のノードCとノードFがバス使用権の要求を発しているノードである。これを受けた親ノード（図10ではノードA）は更に親ノードに向かって、バス使用権の要求を発する（中継する）。この要求は最終的に調停を行なうルートに届けられる。

【0096】

バス使用要求を受けたルートノードは、どのノードにバスを使用させるかを決める。この調停作業はルートノードのみが行なえるものであり、調停によって勝ったノードにはバスの使用許可を与える。図10（b）ではノードCに使用許可が与えられ、ノードFの使用は拒否された図である。

【0097】

アービトレーションに負けたノードに対してはDP（data prefix）パケットを送り、拒否されたことを知らせる。拒否されたノードのバス使用要求は次回のアビトレーションまで待たされる。

【0098】

以上のようにして、アービトレーションに勝ってバスの使用許可を得たノードは、以降データの転送を開始できる。

【0099】

ここで、アービトレーションの一連の流れをフローチャート図11に示して、説明する。

【0100】

ノードがデータ転送を開始できる為には、バスがアイドル状態であることが必要である。先に行われていたデータ転送が終了して、現在バスが空き状態であることを認識するためには、各転送モードで個別に設定されている所定のアイドル時間ギャップ長（例えば、サブアクション・ギャップ）を経過する事によって、各ノードは自分の転送を開始できると判断する。

#### 【0101】

ステップS401として、Async データ、Iso データ等それぞれ転送するデータに応じた所定のギャップ長が得られたか判断する。所定のギャップ長が得られない限り、転送を開始するために必要なバス使用権の要求はできないので、所定のギャップ長が得られるまで待つ。

#### 【0102】

ステップS401で所定のギャップ長が得られたら、ステップS402として転送すべきデータがあるか判断し、ある場合はステップS403として転送するためにバスを確保するよう、バス使用権の要求をルートに対して発する。このときの、バス使用権の要求を表す信号の伝達は、図13に示したように、ネットワーク内各機器を中継しながら、最終的にルートに届けられる。ステップS402で転送するデータがない場合は、そのまま待機する。

#### 【0103】

次に、ステップS404として、ステップS403のバス使用要求を1つ以上ルートが受信したら、ルートはステップS405として使用要求を出したノードの数を調べる。

#### 【0104】

ステップS405での選択値がノード数=1（使用権要求を出したノードは1つ）だったら、そのノードに直後のバス使用許可が与えられることとなる。ステップS405での選択値がノード数>1（使用要求を出したノードは複数）だったら、ルートはステップS406として使用許可を与えるノードを1つに決定する調停作業を行う。この調停作業は公平なものであり、毎回同じノードばかりが許可を得る様なことはなく、平等に権利を与えていくような構成となっている（フェア・アービトレーション）。

【0105】

ステップS407として、ステップS406で使用要求を出した複数ノードの中からルートが調停して使用許可を得た1つのノードと、敗れたその他のノードに分ける選択を行う。ここで、調停されて使用許可を得た1つのノード、またはステップS405の選択値から使用要求ノード数=1で調停無しに使用許可を得たノードには、ステップS408として、ルートはそのノードに対して許可信号を送る。

【0106】

許可信号を得たノードは、受け取った直後に転送すべきデータ（パケット）を転送開始する。また、ステップS406の調停で敗れて、バス使用が許可されなかったノードにはステップS409としてルートから、アービトレーション失敗を示すDP（data prefix）パケットを送られ、これを受け取ったノードは再度転送を行うためのバス使用要求を出すため、ステップS401まで戻り、所定ギャップ長が得られるまで待機する。

【0107】

以上がアービトレーションの流れを説明した、フローチャート図11の説明である。

【0108】

Asynchronous（非同期）転送について説明する。

【0109】

アシンクロナス転送は、非同期転送である。図12にアシンクロナス転送における時間的な遷移状態を示す。

【0110】

図12の最初のサブアクション・ギャップは、バスのアイドル状態を示すものである。このアイドル時間が一定値になった時点で、転送を希望するノードはバスが使用できると判断して、バス獲得のためのアービトレーションを実行する。

【0111】

アービトレーションでバスの使用許可を得ると、次にデータの転送がパケット形式で実行される。データ転送後、受信したノードは転送されたデータに対して

の受信結果のack（受信確認用返送コード）をack gap という短いギャップの後、返送して応答するか、応答パケットを送ることによって転送が完了する。ack は4ビットの情報と4ビットのチェックサムからなり、成功か、ビジー状態か、ペンディング状態であるかといった情報を含み、すぐに送信元ノードに返送される。

【0112】

つぎに、図13にアシンクロナス転送のパケットフォーマットの例を示す。

【0113】

パケットには、データ部及び誤り訂正用のデータCRCの他にはヘッダ部があり、そのヘッダ部には図13に示したような、目的ノードID、ソースノードID、転送データ長さや各種コードなどが書き込まれ、転送が行なわれる。

【0114】

また、アシンクロナス転送は自己ノードから相手ノードへの1対1の通信である。転送元ノードから転送されたパケットは、ネットワーク中の各ノードに行き渡るが、自分宛てのアドレス以外のものは無視されるので、宛先の1つのノードのみが読込むことになる。

【0115】

以上がアシンクロナス転送の説明である。

【0116】

Isochronous（同期）転送について説明する。

【0117】

アイソクロナス転送は同期転送である。1394シリアルバスの最大の特徴であるともいえるこのアイソクロナス転送は、特に映像データや音声データといったマルチメディアデータなど、リアルタイムな転送を必要とするデータの転送に適した転送モードである。

【0118】

また、アシンクロナス転送（非同期）が1対1の転送であったのに対し、このアイソクロナス転送はブロードキャスト機能によって、転送元の1つのノードから他のすべてのノードへ一様に転送される。

## 【0119】

図14はアイソクロナス転送における、時間的な遷移状態を示す図である。

## 【0120】

アイソクロナス転送は、バス上一定時間毎に実行される。この時間間隔をアイソクロナスサイクルと呼ぶ。アイソクロナスサイクル時間は、 $125\mu\text{S}$ である。この各サイクルの開始時間を示し、各ノードの時間調整を行なう役割を担っているのがサイクル・スタート・パケットである。サイクル・スタート・パケットを送信するのは、サイクル・マスタと呼ばれるノードであり、1つ前のサイクル内の転送終了後、所定のアイドル期間（サブアクションギャップ）を経た後、本サイクルの開始を告げるサイクル・スタート・パケットを送信する。

## 【0121】

このサイクル・スタート・パケットの送信される時間間隔が $125\mu\text{S}$ となる。

## 【0122】

また、図14にチャネルA、チャネルB、チャネルCと示したように、1サイクル内において複数種のパケットがチャネルIDをそれぞれ与えられることによって、区別して転送できる。これによって同時に複数ノード間でのリアルタイムな転送が可能であり、また受信するノードでは自分が欲しいチャネルIDのデータのみを取り込む。このチャネルIDは送信先のアドレスを表すものではなく、データに対する論理的な番号を与えているに過ぎない。よって、あるパケットの送信は1つの送信元ノードから他のすべてのノードに行き渡る、ブロードキャストで転送されることになる。

## 【0123】

アイソクロナス転送のパケット送信に先立って、アシンクロナス転送同様アービトラションが行われる。しかし、アシンクロナス転送のように1対1の通信ではないので、アイソクロナス転送にはack（受信確認用返信コード）は存在しない。

## 【0124】

また、図14に示した iso gap（アイソクロナスギャップ）とは、アイソクロ

ナス転送を行なう前にバスが空き状態であると認識するために必要なアイドル期間を表している。この所定のアイドル期間を経過すると、アイソクロナス転送を行ないたいノードはバスが空いていると判断し、転送前のアービトレーションを行なうことができる。

## 【0125】

つぎに、図15にアイソクロナス転送のフォーマットの例を示し、説明する。

## 【0126】

各チャンネルに分かれた、各種のパケットにはそれぞれデータ部及び誤り訂正用のデータCRCの他にヘッダ部があり、そのヘッダ部には図17に示したような、転送データ長やチャンネルNO、その他各種コード及び誤り訂正用のヘッダCRCなどが書き込まれ、転送が行なわれる。

## 【0127】

以上がアイソクロナス転送の説明である。

## 【0128】

バス・サイクルについて説明する。

## 【0129】

実際の1394シリアルバス上の転送では、アイソクロナス転送と、アシンクロナス転送は混在できる。その時の、アイソクロナス転送とアシンクロナス転送が混在した、バス上の転送状態の時間的な遷移の様子を表した図を図16に示す。

## 【0130】

アイソクロナス転送はアシンクロナス転送より優先して実行される。その理由は、サイクル・スタート・パケットの後、アシンクロナス転送を起動するために必要なアイドル期間のギャップ長（サブアクションギャップ）よりも短いギャップ長（アイソクロナスギャップ）で、アイソクロナス転送を起動できるからである。したがって、アシンクロナス転送より、アイソクロナス転送は優先して実行されることとなる。

## 【0131】

図16に示した、一般的なバスサイクルにおいて、サイクル#mのスタート時にサイクル・スタート・パケットがサイクル・マスタから各ノードに転送される。これによって、各ノードで時刻調整を行ない、所定のアイドル期間（アイソクロナスギャップ）を待ってからアイソクロナス転送を行なうべきノードはアービトレーションを行い、パケット転送に入る。図16ではチャンネルeとチャンネルsとチャンネルkが順にアイソクロナス転送されている。

【0132】

このアービトレーションからパケット転送までの動作を、与えられているチャンネル分繰り返し行なった後、サイクル#mにおけるアイソクロナス転送がすべて終了したら、アシンクロナス転送を行うことができるようになる。

【0133】

アイドル時間がアシンクロナス転送が可能なサブアクションギャップに達する事によって、アシンクロナス転送を行いたいノードはアービトレーションの実行に移れると判断する。

【0134】

ただし、アシンクロナス転送が行える期間は、アイソクロナス転送終了後から、次のサイクル・スタート・パケットを転送すべき時間（cycle synch）までの間にアシンクロナス転送を起動するためのサブアクションギャップが得られた場合に限っている。

【0135】

図16のサイクル#mでは3つのチャンネル分のアイソクロナス転送と、その後アシンクロナス転送（含むack）が2パケット（パケット1、パケット2）転送されている。このアシンクロナスパケット2の後は、サイクルm+1をスタートすべき時間（cycle synch）にいたるので、サイクル#mでの転送はここまでで終わる。

【0136】

ただし、非同期または同期転送動作中に次のサイクル・スタート・パケットを送信すべき時間（cycle synch）に至ったとしたら、無理に中断せず、その転送が終了した後のアイドル期間を待ってから次サイクルのサイクル・スタート・パ



ケットを送信する。すなわち、1つのサイクルが $125\mu\text{S}$ 以上続いたときは、その分次サイクルは基準の $125\mu\text{S}$ より短縮されたとする。このようにアイソクロナス・サイクルは $125\mu\text{S}$ を基準に超過、短縮し得るものである。

【0137】

しかし、アイソクロナス転送はリアルタイム転送を維持するために毎サイクル必要であれば必ず実行され、アシンクロナス転送はサイクル時間が短縮されたことによって次以降のサイクルにまわされることもある。

こういった遅延情報も含めて、サイクル・マスタによって管理される。

【0138】

そこで、まず、第1の実施の形態について説明する。

【0139】

図17は、本発明の特徴を最もよく表す図であり、同図において1394のインターフェイスをLANで用いられるOSIモデルの各層と対比させてみると、OSIモデルの物理層1とデータリンク層2が、1394インターフェイスの下位層4であるフィジカル・リンク層に該当し、それら下位層4の上に存在する上位層3が1394インターフェイスではトランスポートプロトコル層5とプレゼンテーション層6に該当する。また、本発明の特徴となるLOGINプロトコル7は、1394インターフェイスの下位層4とトランスポートプロトコル5との間で動作するものである。

【0140】

この実施の形態では、シリアルバスプロトコル(SBP-2)8に準拠したデバイスにLOGINプロトコルを挿入することによって、相手のデバイスに対して自分がSBP-2に準拠したプロトコルを使ってやり取りを行いたいことを通知することができる。また例2では、1394インターフェイス上で特化されたデバイスプロトコル9についてもLOGINプロトコルを挿入することで、お互いのデバイスがそのプロトコルがサポートされているかを判別してデータのやり取りを行うことができる。

【0141】

図18は、LOGINプロトコルの基本動作を示した図であり、プリンタは印

字タスク10を実行する際にまず初めにLOGINプロトコルを使ってプリンタで用意しているプリンタプロトコルA・B・Cのうち、どれを選択して印字するかを決定し、その後は決定されたプロトコルに沿って印字動作を行う。

すなわち、プリンタ側でいくつかのプリンタプロトコルをサポートしているデバイスにおいて、ターゲットとの接続の際にまず初めに相手のデバイスのプロトコルをLOGINプロトコルを使って判別し、プリンタは相手のプロトコルに合わせたプリンタプロトコルを複数の中からひとつ選択し、この選ばれたプロトコルに沿って印字データやコマンドのやり取りを行って印字処理を行う。

#### 【0142】

図19は、この実施の形態におけるLOGINプロトコルを実装した1394インターフェイスにおける各デバイスの接続形態を示した図で、複数のプリンタプロトコルに対応したプリンタ11に対してLOGINプロトコルを実装したデバイス（PC12、スキャナ13、VCR14等）が接続された場合に、LOGINプロトコルを使用して相手のトランスポートプロトコルに応じてプリンタ側でプリンタプロトコルを切り替えることにより、各デバイスからの印字タスクを問題なく処理することが可能となる。

#### 【0143】

図20はログイン動作の流れを示したものである。

#### 【0144】

第一ステップ：

- ・デバイス（この場合マルチプロトコルプリンタ）をロック
- ・プリンタのケーパビリティ（受け付けるプロトコル等）を取得  
かかるケーパビリティは、後述するレジスタ503に格納される。
- ・ホストのケーパビリティをプリンタにセット

第二ステップ：

- ・第一ステップで決定したプロトコルで、プリントデータの通信

第三ステップ：

- ・プリンタとホストは、コネクションを切断

#### 【0145】

図21は、この実施の形態においてLOGINプロトコルのためにプリンタが備える1394シリアルバスのCSRを示し、図中、501はロックレジスタ(lock)、502はプロトコルレジスタ(protocol)、503はケーパビリティレジスタ(capability)を示す。これらのレジスタは1394シリアルバスのアドレス空間における初期ユニット空間の定められたアドレスに配置される。ロックレジスタ501はリソースのロック状態を示し、値0はログイン可能な状態をあらわし、0以外はロック状態ですでにログインされていることをあらわす。ケーパビリティレジスタ503は設定可能なプロトコルを示し、ビット毎にプロトコルに対応し、ビットの値1はそのプロトコルが設定可能であることをあらわし、0は設定不可能であることをあらわす。プロトコルレジスタ502は現在設定されたプロトコルを示し、設定されたプロトコルに対応するケーパビリティレジスタ503のビットに相当するビットの値が1になる。

【0146】

図22は、ログイン処理のホスト側のフローチャートである。

【0147】

ログインを開始するためには、まずログインしようとするプリンタのロックレジスタ501、プロトコルレジスタ502、ケーパビリティレジスタ503をリードトランザクションにより確認する。ここで、ケーパビリティレジスタ503の内容から、ホストが通信に用いるプロトコルをプリンタがサポートしているかどうか確認する(ステップ601)。もしホストのプロトコルがプリンタのサポート外ならば、次のステップでログインを中止する。

【0148】

ロックレジスタ501が0以外であれば、他のデバイスがログイン中であるとみなし、ログインを中止する。ログインレジスタ501が0であれば、現在ログイン可能とみなす(ステップ602)。

【0149】

ログイン可能な場合リソースロック処理に移り、プリンタのロックレジスタ501にロックトランザクションを用いて1を書き込み、ホスト側のログイン設定とする(ステップ603)。この状態でプリンタはロックされたことになり、他

のデバイスからの出力は不可能。またレジスタの変更も不可能とする。

【0150】

上記のようにプリンタのリソースがロックされた状態で、次にプロトコルの設定を行なう。本実施の形態におけるプリンタは、複数プリントプロトコルをサポートするため、プリントデータを受け取る前に、ホスト側のプロトコルを知らねばならない。ここでは、ホスト側がプリンタのプロトコルレジスタ502に対して、ライトランザクションにより、これから使用するプロトコルに相当するビットを立て通知する（ステップ604）、という手段を用いる。

【0151】

この時点で、ホストが通信に用いるプロトコルがプリンタに通知され、かつプリンタがロック状態なので、現在ログインしているホストが通常のプロトコルでプリントデータの送信を行なう（ステップ605）。

【0152】

プリントデータの送信が終了したら、ホストはプリンタのロックレジスタ501、プロトコルレジスタ502、ケーパビリティレジスタ503をクリアすることにより、プリンタからログアウトする（ステップ606）。

【0153】

図23は、ログイン処理のプリンタ側のフローチャートである。

【0154】

プリンタ側は、通常ホストからのログイン待ちの状態にいる。ホストからのプリントリクエストは、プリンタのロックレジスタ501、プロトコルレジスタ502、ケーパビリティレジスタ503の読み取りにより開始されるので、前記レジスタは、常に他のデバイスから読み出し可能の状態にしておく。今、プリントアウトを実行しようとするホストがプリンタをロックしたとする（ステップ701）。

【0155】

ロックレジスタ501への書き込みによりプリンタがロックされたら、次にホストからの、使用プロトコルの通知を待つ（ステップ702）。ロック状態になってからプロトコルの通知を待つのは、ログインの途中で、さらに他のデバイス

からのリクエストにより、プロトコルレジスタ502を書き換えられないようにするためである。

【0156】

プロトコルの通知があったら、プリンタは自分の受け付けるプロトコルをスイッチし、ホスト側に合わせ、通信を行なう（ステップ703～710）。

【0157】

通信が終了したら、プリンタはロックレジスタ501、プロトコルレジスタ502、ケーパビリティレジスタ503がクリアされたのを確認し（ステップ711）、ログイン待ち状態（ステップ701）に戻る。

【0158】

つぎに、第2の実施の形態について説明する。

【0159】

図24は、この実施の形態における動作を示した図であり、図2に示した第1の実施の形態と比較してみるとLOGINプロトコルを実装していないプロトコルDについても対応している点が特徴といえる。

すなわち、LOGINプロトコルを持ったターゲットに対してだけでなく、既存のプロトコルD（例えばAV/Cプロトコル）にのみ対応しているデバイスに対しても印字動作の保証を行う為に、プリンタ側で対応するプリンタプロトコルをLOGINプロトコルの持っていないデバイス用に追加を行ったものである。

【0160】

この動作について説明すると、接続の初めに行われるLOGINプロトコルによる会話によってターゲット側がLOGINプロトコルに対応していないことをプリンタ側が認識した際、プリンタ側はその他のプロトコルDを使ってターゲットに会話を行い、そこで会話が成立した場合はそのプロトコルDに沿った印字タスクを実行することができる。

【0161】

図25は、この実施の形態におけるOSIモデルとの対比を示した図であり、この実施の形態ではLOGINプロトコルの実装されていない現行のAV/Cプロトコルに準拠したプリンタをモデルとしている。例4は、現行のプロトコルで

はまだ定義されていない、スキャナ用に作られたLOGINプロトコルの機能を持っていないプロトコルが実装されていた場合を表している。

すなわち、LOGINプロトコルを実装していないプロトコルについてもプリンタ側で対応することができれば、繋がるターゲットデバイスの種類に幅ができるわけである。

【0162】

尚、本発明は、図19に示すような、複数の機器から構成されるシステムに適用しても、1つの機器からなる装置内のデータ処理方法に適用してもよい。

【0163】

また、本発明の目的は、上述した各実施の形態のホスト及び端末の機能を実現するソフトウェアのプログラムコードを記憶した記憶媒体を、システム或いは装置に供給し、そのシステム或いは装置のコンピュータ（又はCPUやMPU）が記憶媒体に格納されたプログラムコードを読みだして実行することによっても、達成されることは言うまでもない。

【0164】

この場合、記憶媒体から読み出されたプログラムコード自体が前述した各実施の形態の機能を実現することとなり、そのプログラムコードを記憶した記憶媒体は本発明を構成することとなる。

【0165】

プログラムコードを供給するための記憶媒体としては、例えば、フロッピーディスク、ハードディスク、光ディスク、光磁気ディスク、CD-ROM、CD-R、磁気テープ、不揮発性のメモリカード、ROM等を用いることができる。

【0166】

また、コンピュータが読みだしたプログラムコードを実行することにより、前述した実施の形態の機能が実現されるだけでなく、そのプログラムコードの指示に基づき、コンピュータ上で稼動しているOS等が実際の処理の一部又は全部を行い、その処理によって実施の形態の機能が実現される場合も含まれることは言うまでもない。

【0167】

さらに、記憶媒体から読み出されたプログラムコードが、コンピュータに挿入された拡張機能ボードやコンピュータに接続された機能拡張ユニットに備わるメモリに書き込まれた後、そのプログラムコードの指示に基づき、その機能拡張ボードや機能拡張ユニットに備わるCPUなどが実際の処理の一部又は全部を行い、その処理によって前述した実施の形態の機能が実現される場合も含まれることは言うまでもない。

【0168】

【発明の効果】

以上説明したように本発明によれば、拡張性の高いデータ処理方法、データ処理装置、プリンタ及び記憶媒体を提供することができる。特に、複数種類のプリンタのプロトコルに対応可能であるので、拡張性が極めて高い。

【図面の簡単な説明】

【図1】

IEEE1394ケーブルを用いた通信システムの一例を示す図である。

【図2】

IEEE1394の階層構造を示す図である。

【図3】

IEEE1394のアドレスを示す図である。

【図4】

IEEE1394のケーブル断面図である。

【図5】

DS-Link 符号化方式を説明するための図である。

【図6】

バスリセットからIDの設定までを説明するフローチャートである。

【図7】

ルートの決定方法を説明するフローチャートである。

【図8】

親子関係決定からすべてのノードIDの設定までの手順を説明するフローチャートである。

【図9】

ノード間の親子関係を示す図である。

【図10】

アービトレーションの過程を説明するための図である。

【図11】

アービトレーションの過程を示すフローチャートである。

【図12】

Asynchronous転送におけるサブアクションを示す図である。

【図13】

Asynchronous転送におけるパケット構造を示す図である。

【図14】

Isochronous 転送におけるサブアクションを示す図である。

【図15】

Isochronous 転送におけるパケット構造を示す図である。

【図16】

IEEE 1394の通信サイクルの一例を説明するための図である。

【図17】

本発明を適用したシステムを説明するための図である。

【図18】

LOGINプロトコルの基本動作を説明するための図である。

【図19】

LOGINプロトコルを実装した1394インターフェイスにおける各デバイスの接続形態を説明するための図である。

【図20】

ログイン動作の流れを説明するための図である。

【図21】

LOGINプロトコルのためにプリンタが備える1394シリアルバスのCSRを説明するための図である。

【図22】



ログイン処理のホスト側の処理を説明するためのフローチャートである。

【図 2 3】

ログイン処理のプリンタ側の処理を説明するためのフローチャートである。

【図 2 4】

上記システムにおける動作を説明するための図である。

【図 2 5】

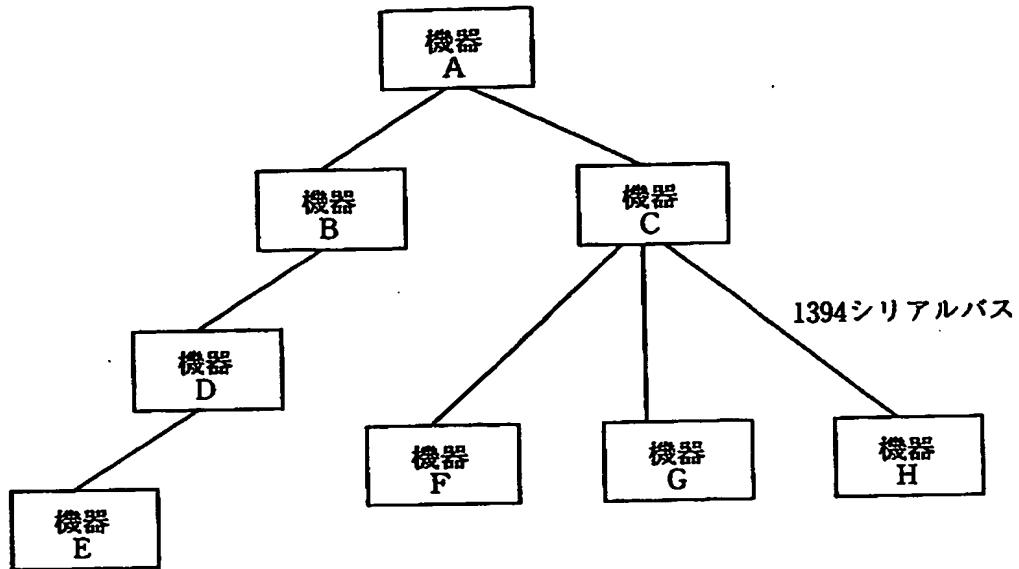
OSIモデルとの対比を説明するための図である。

【符号の説明】

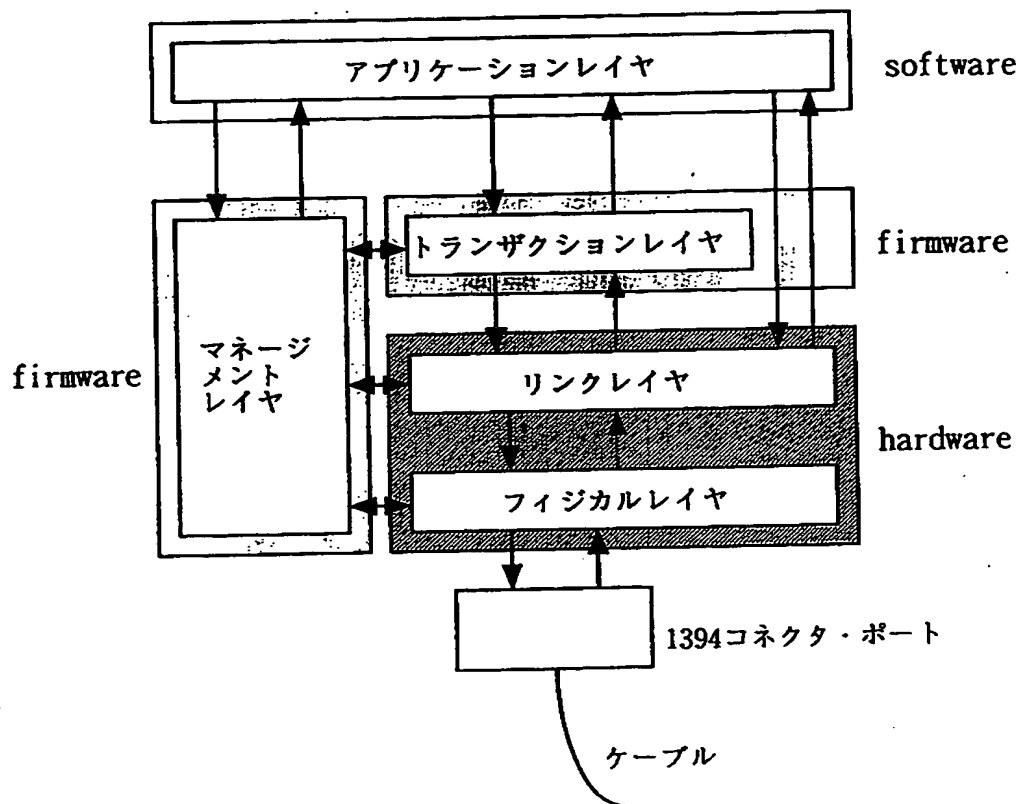
- 1 OSIモデルの物理層
- 2 データリンク層
- 3 上位層
- 4 上位層
- 5 トラnsポートプロトコル層
- 6 プレゼンテーション層
- 7 LOGINプロトコル
- 8 シリアルバスプロトコル (SBP-2)
- 9 デバイスプロトコル

【書類名】 図面

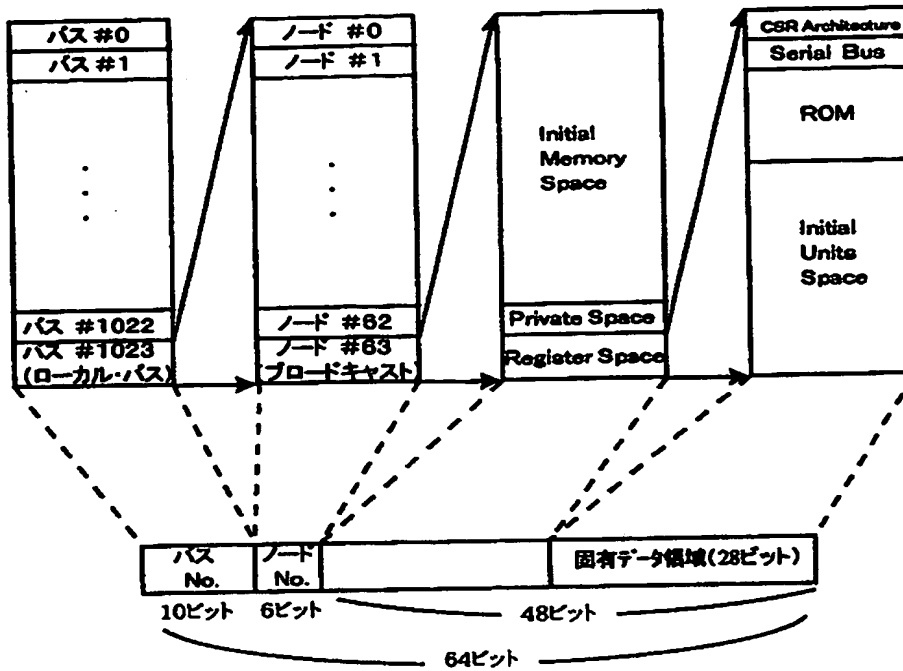
【図1】



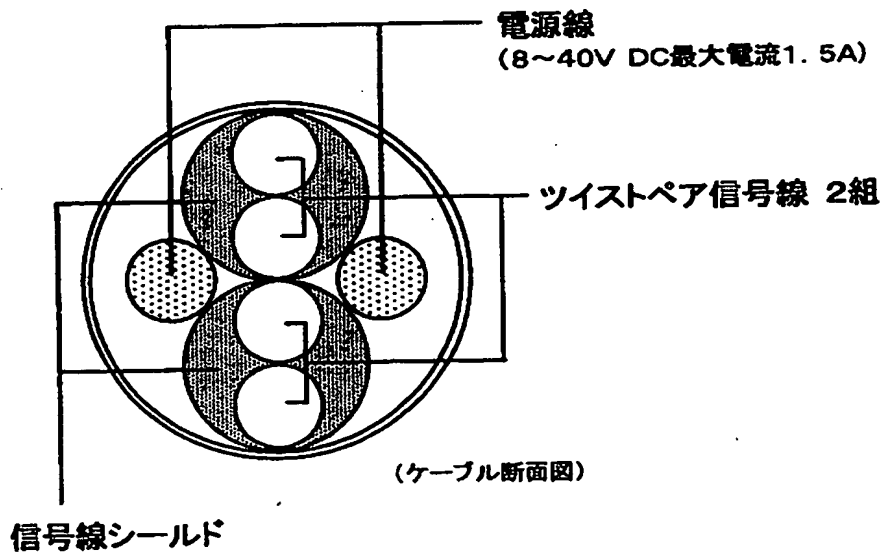
【図2】



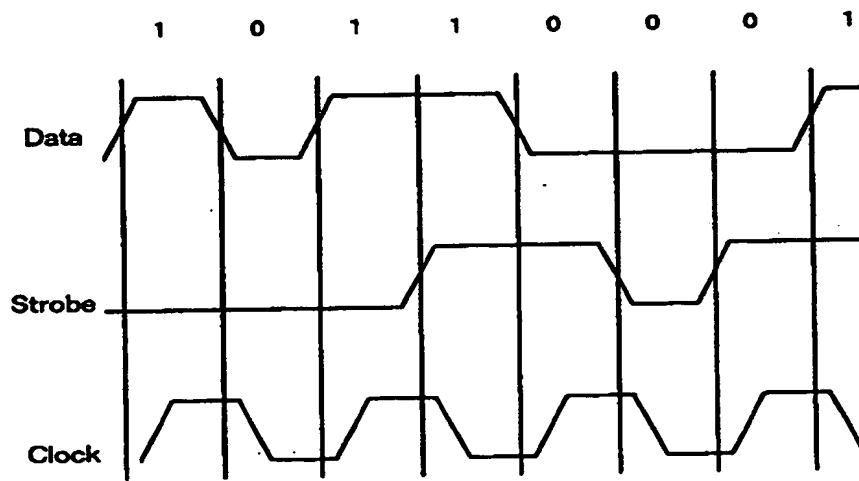
【図3】



【図4】

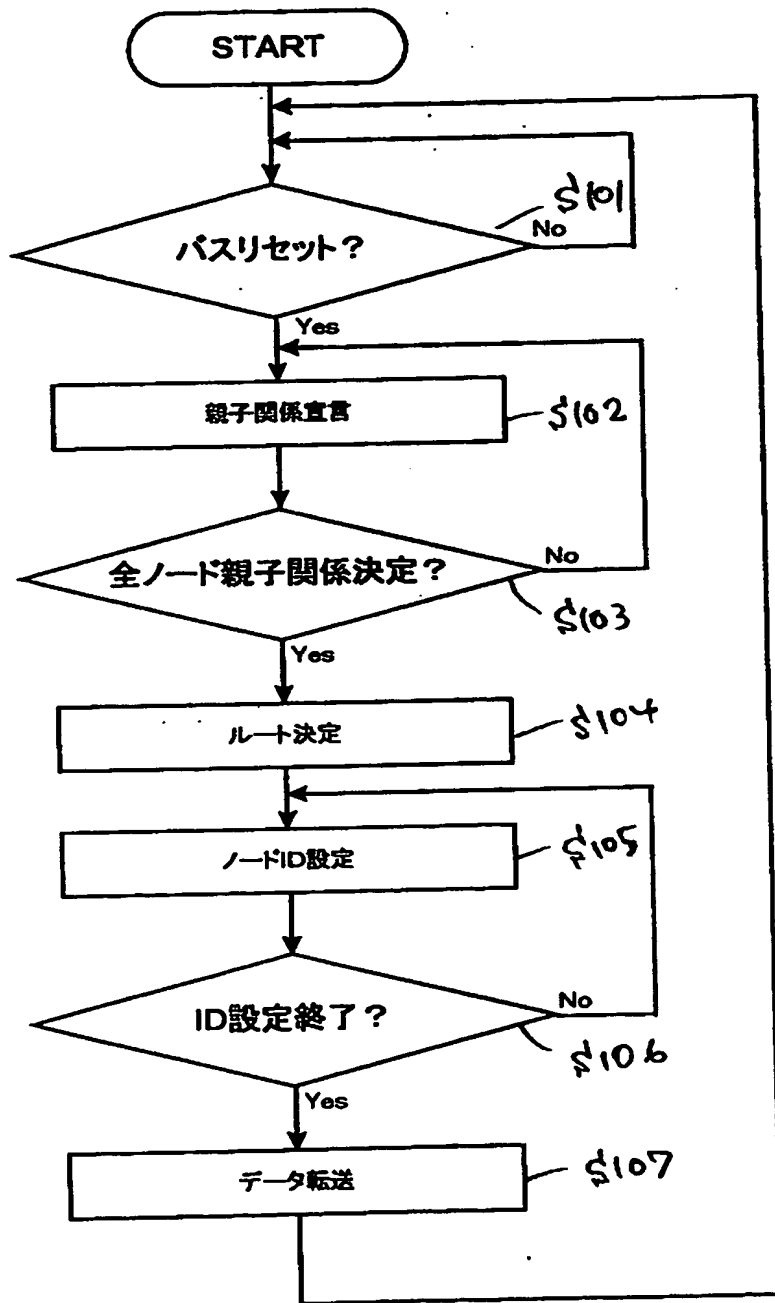


【図5】

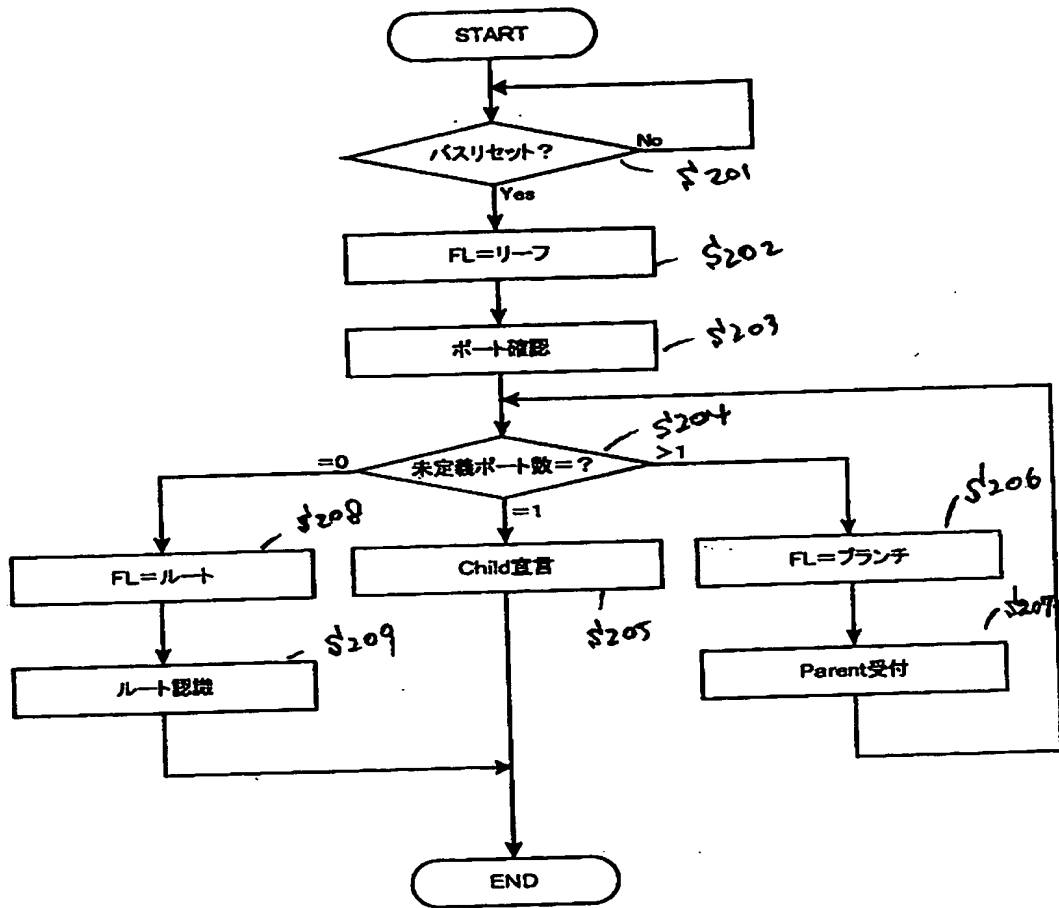


(DataとStrobeの排他的論理和信号)

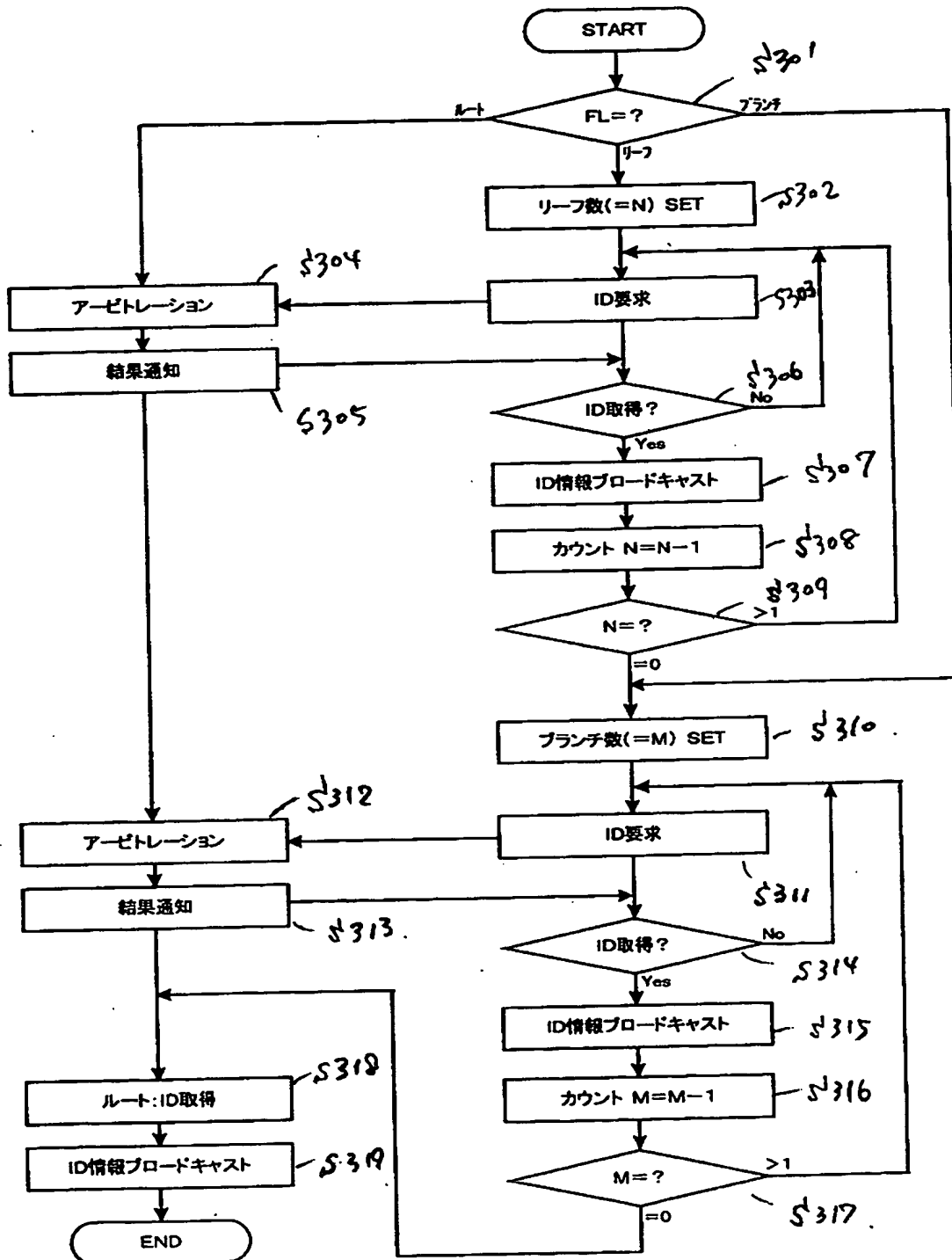
【図6】



【図7】



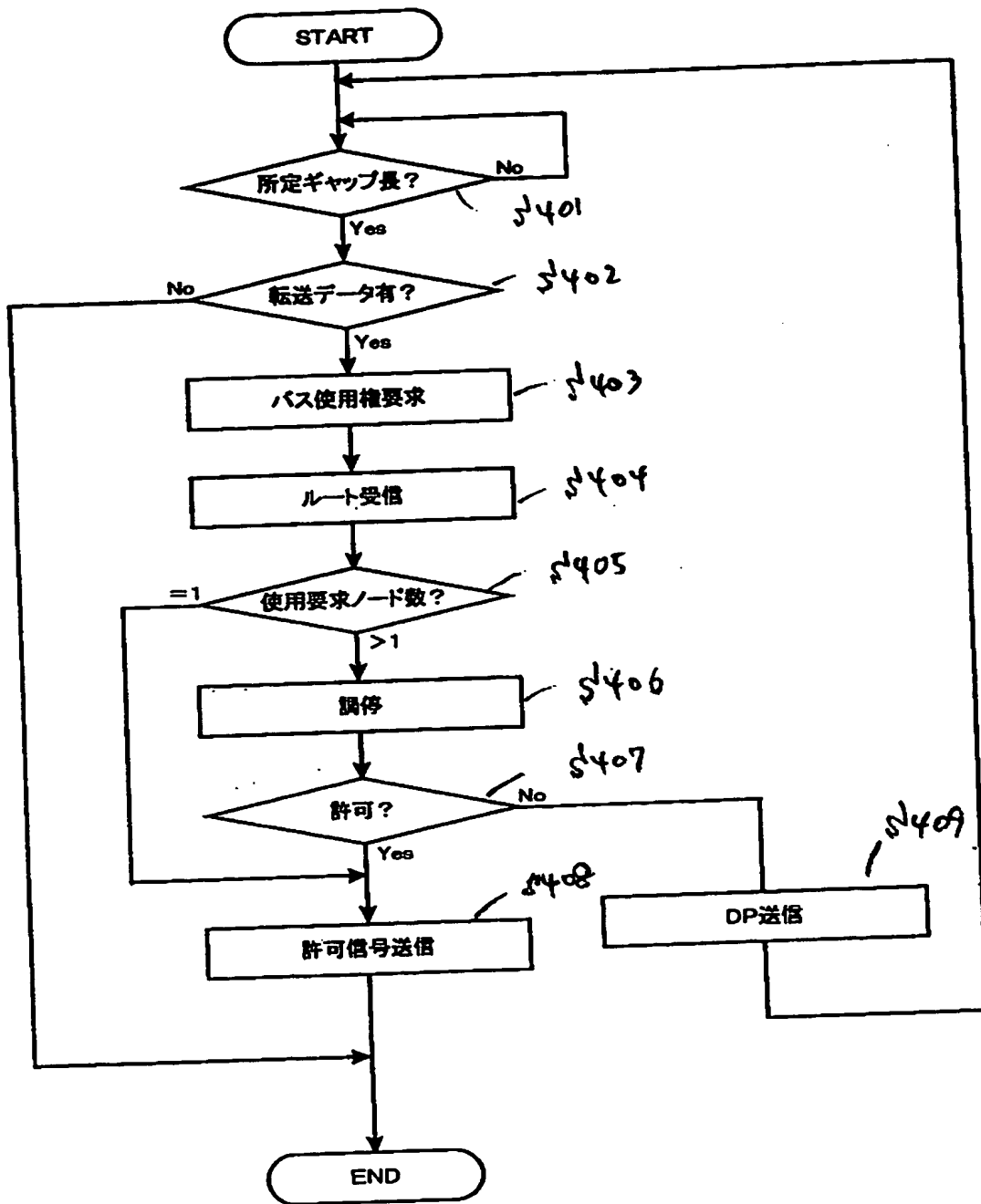
【図 8】



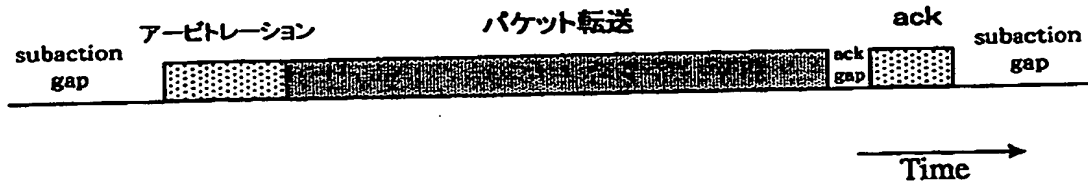




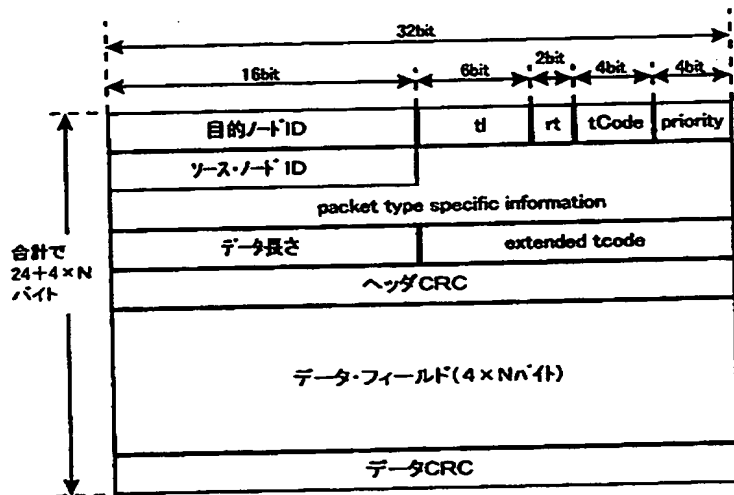
【図11】



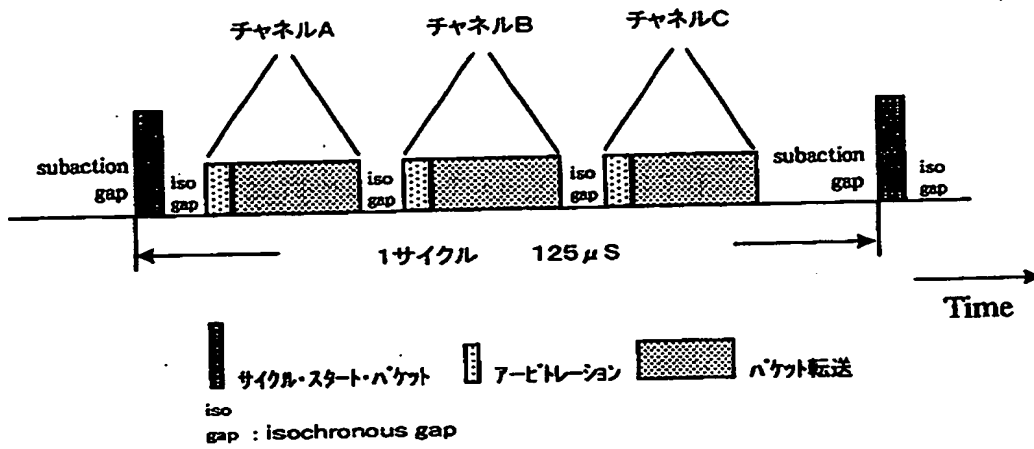
【図12】



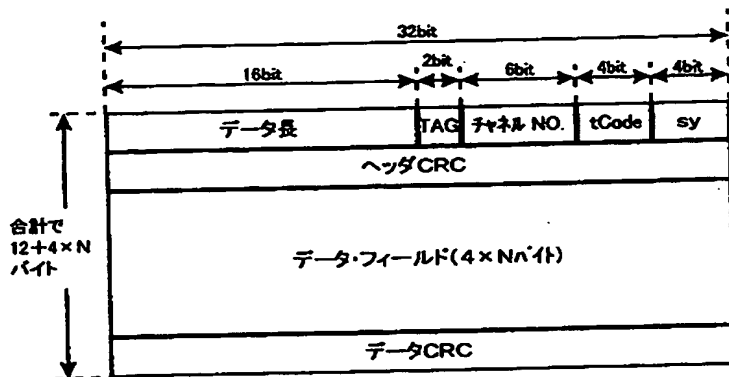
【図13】



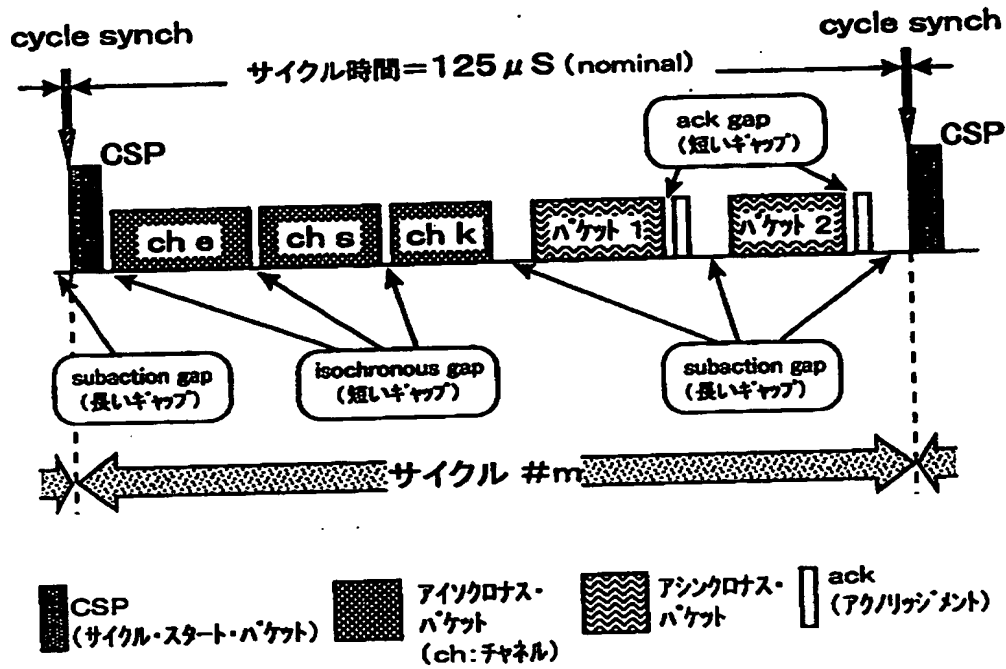
【図14】



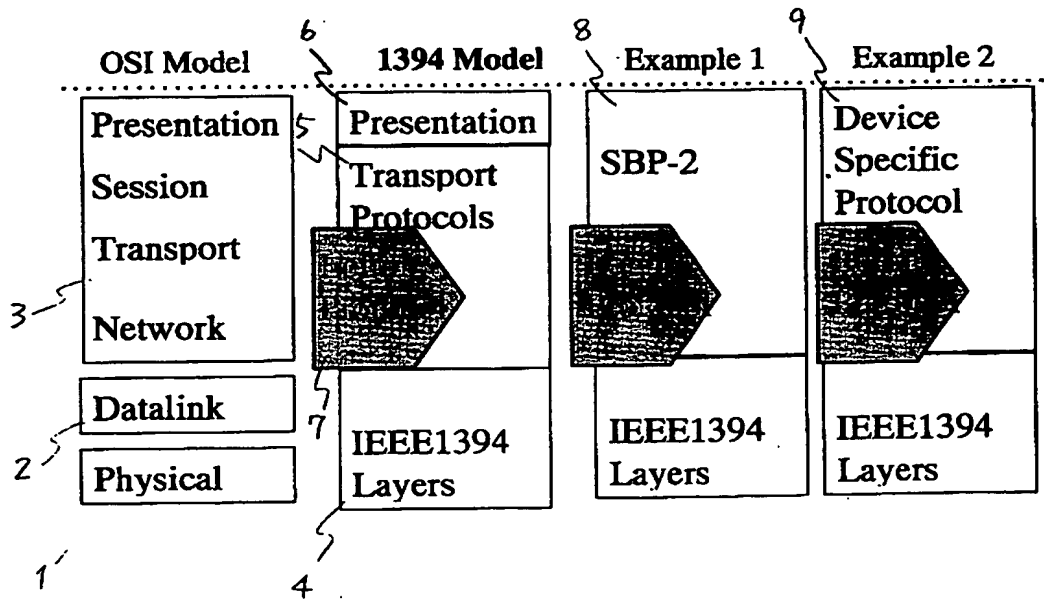
【図15】



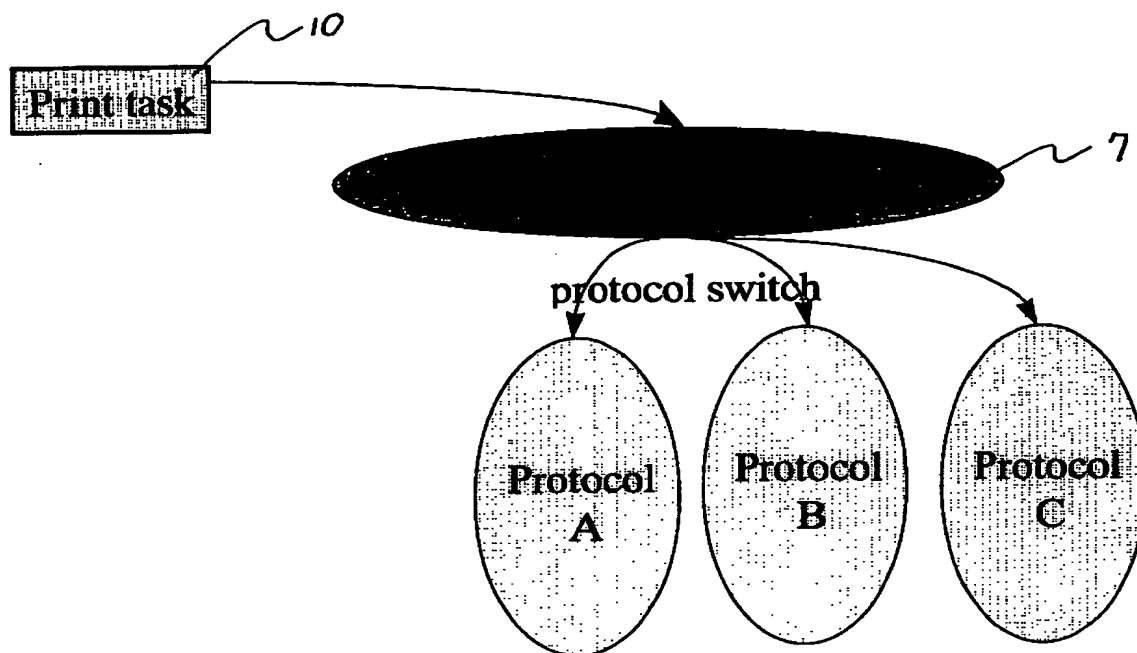
【図 1 6】



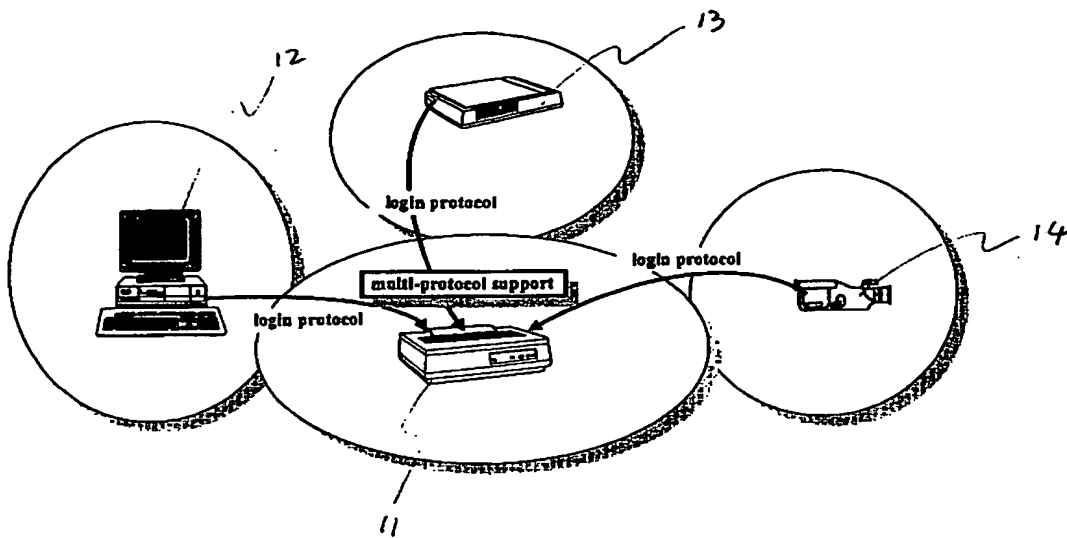
【図 1 7】



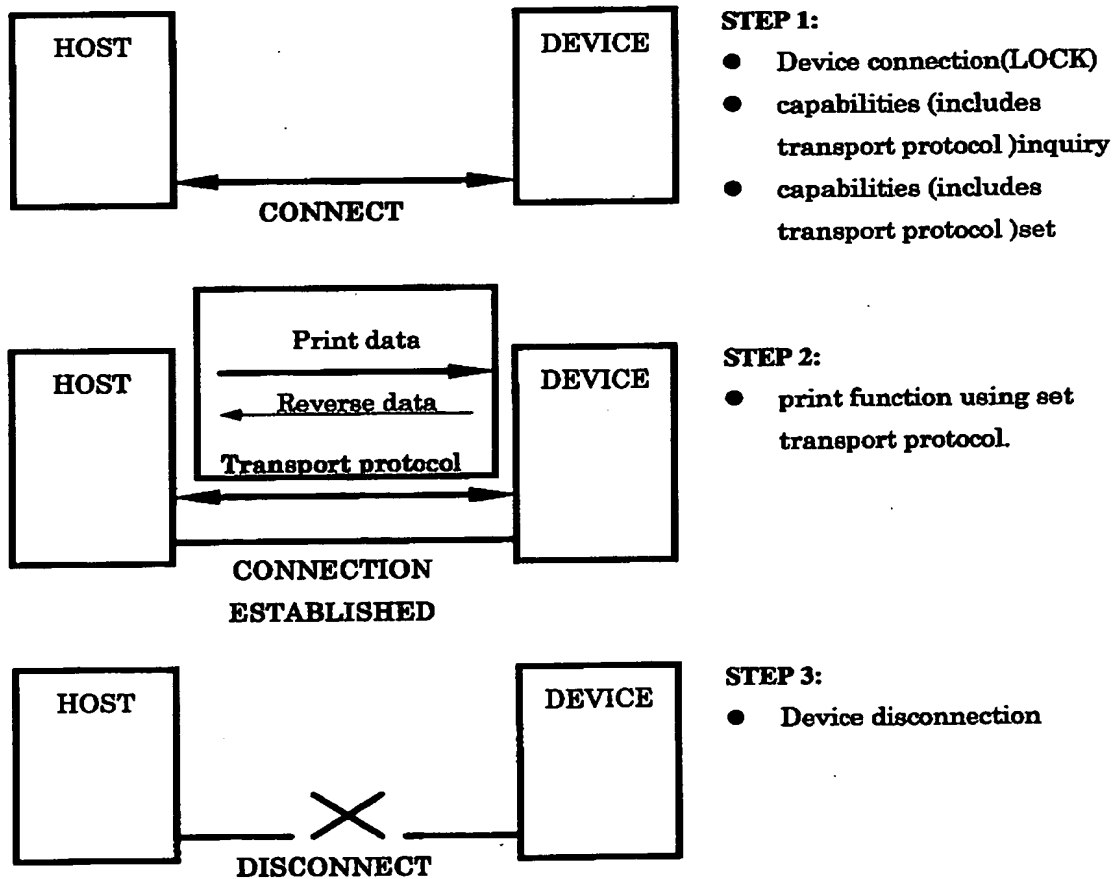
【図 18】



【図 19】



【図 20】



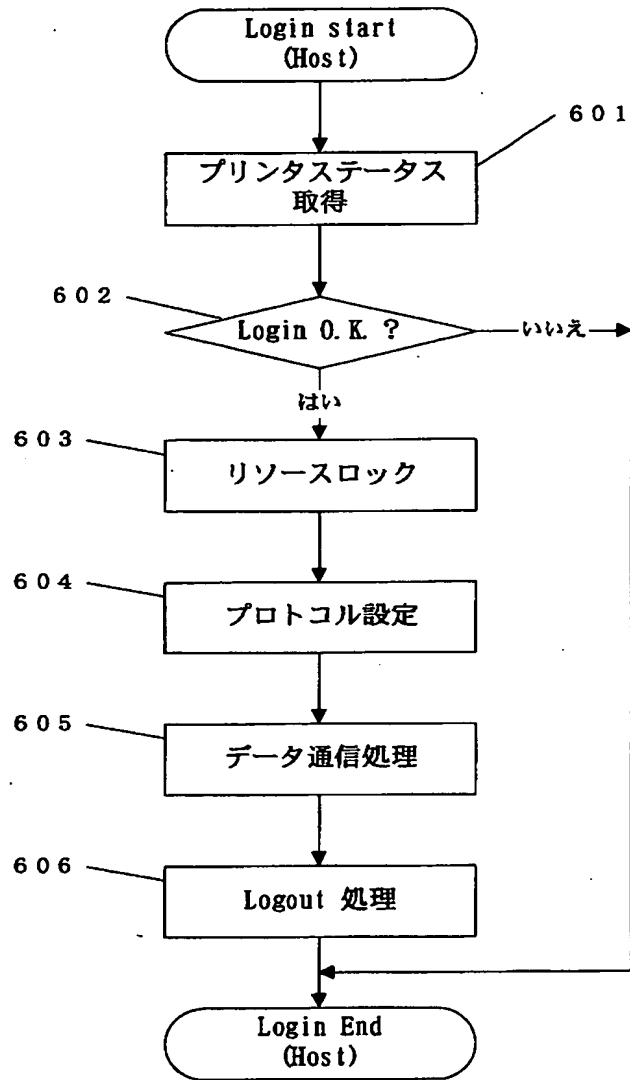


特平 9-030982

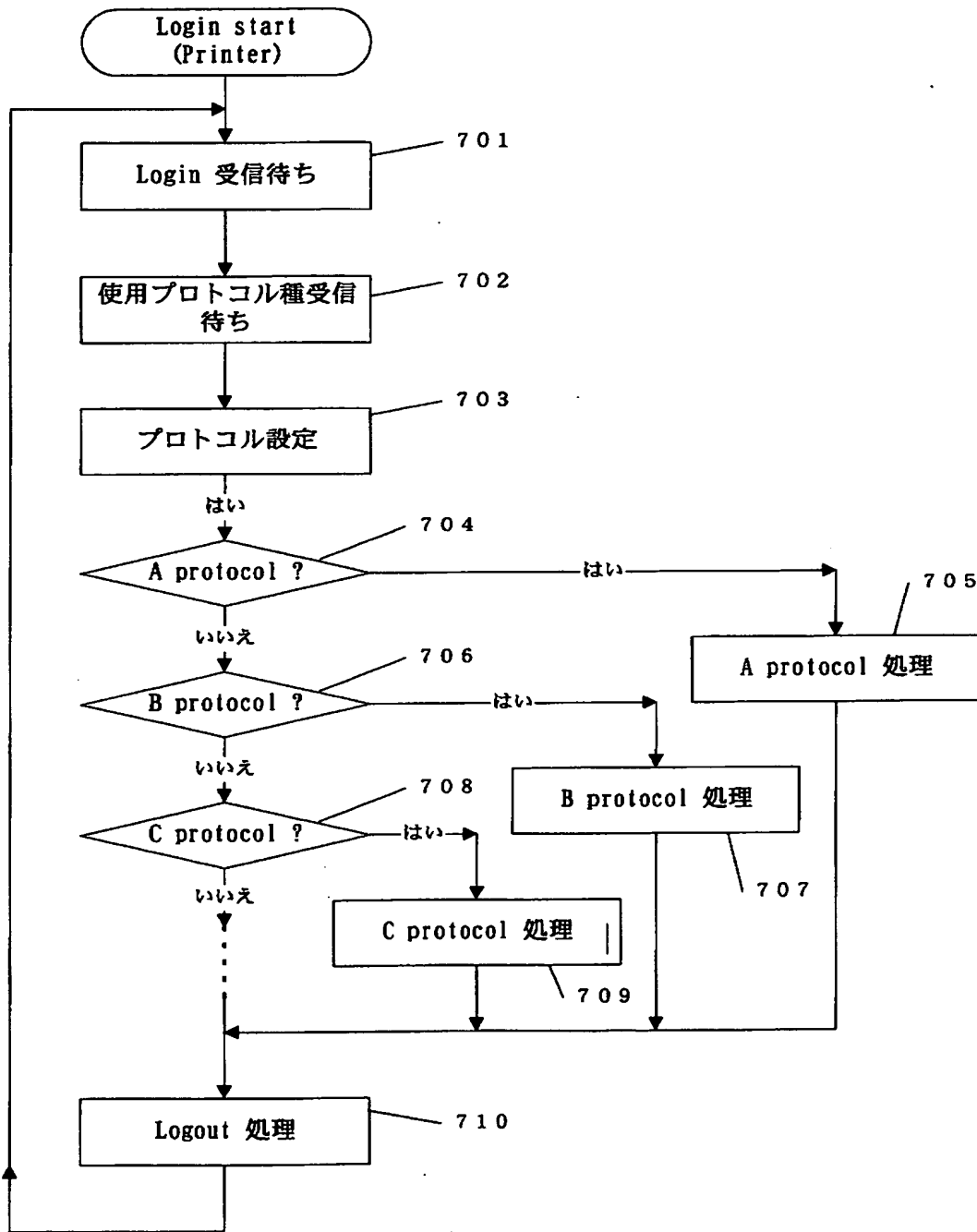
【図 2 1】



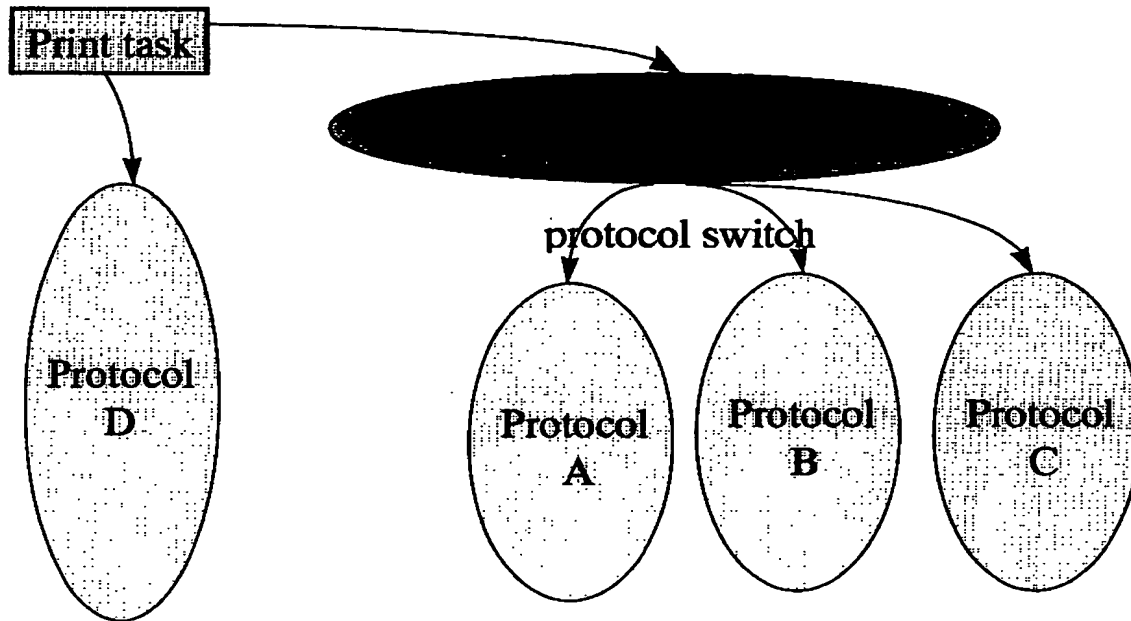
【図22】



【図23】

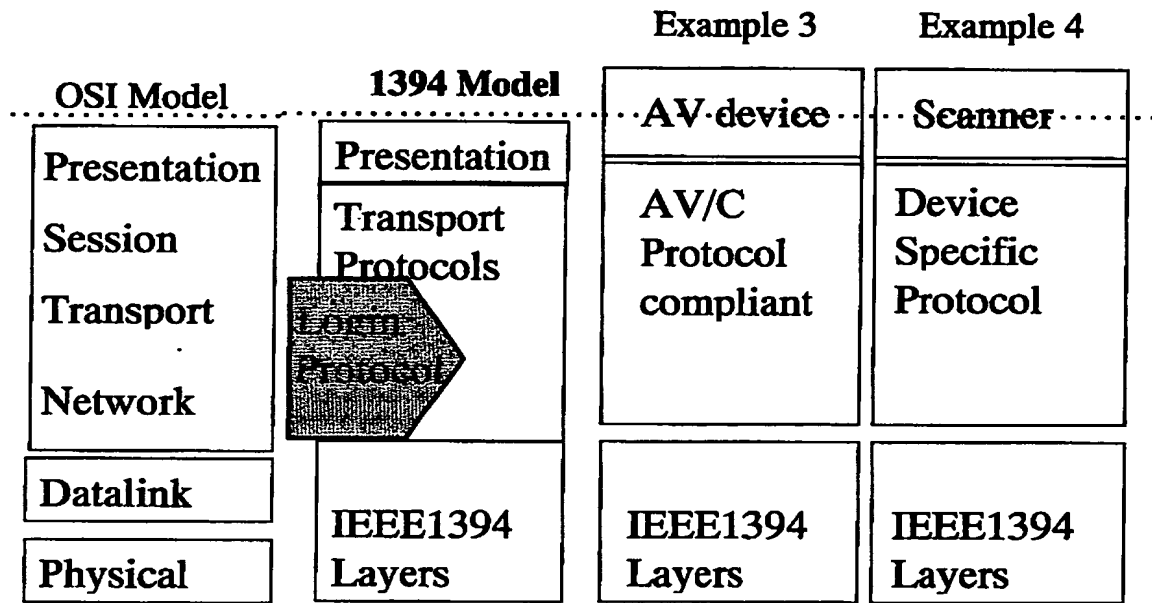


【図24】



特平 9-030982

【図25】



【書類名】 要約書

【要約】

【課題】 システムバスを介してプリントデータを送出する際に、拡張性の高いデータ処理方法を提供する。

【解決手段】 プリンタのプロトコルの種類に係わらない初期プロトコルを実行し、その初期プロトコルに続いて、プリンタ固有のプロトコルを実行する。

【選択図】 図1



【書類名】 職権訂正データ  
【訂正書類】 特許願

<認定情報・付加情報>

【特許出願人】  
【識別番号】 000001007  
【住所又は居所】 東京都大田区下丸子3丁目30番2号  
【氏名又は名称】 キヤノン株式会社  
【代理人】 申請人  
【識別番号】 100090273  
【住所又は居所】 東京都豊島区東池袋1丁目17番8号 池袋TGホ  
ーメストビル5階 國分特許事務所  
【氏名又は名称】 國分 孝悦

出 願 人 履 歴 情 報

識別番号 [000001007]

1. 変更年月日 1990年 8月30日  
[変更理由] 新規登録  
住 所 東京都大田区下丸子3丁目30番2号  
氏 名 キヤノン株式会社